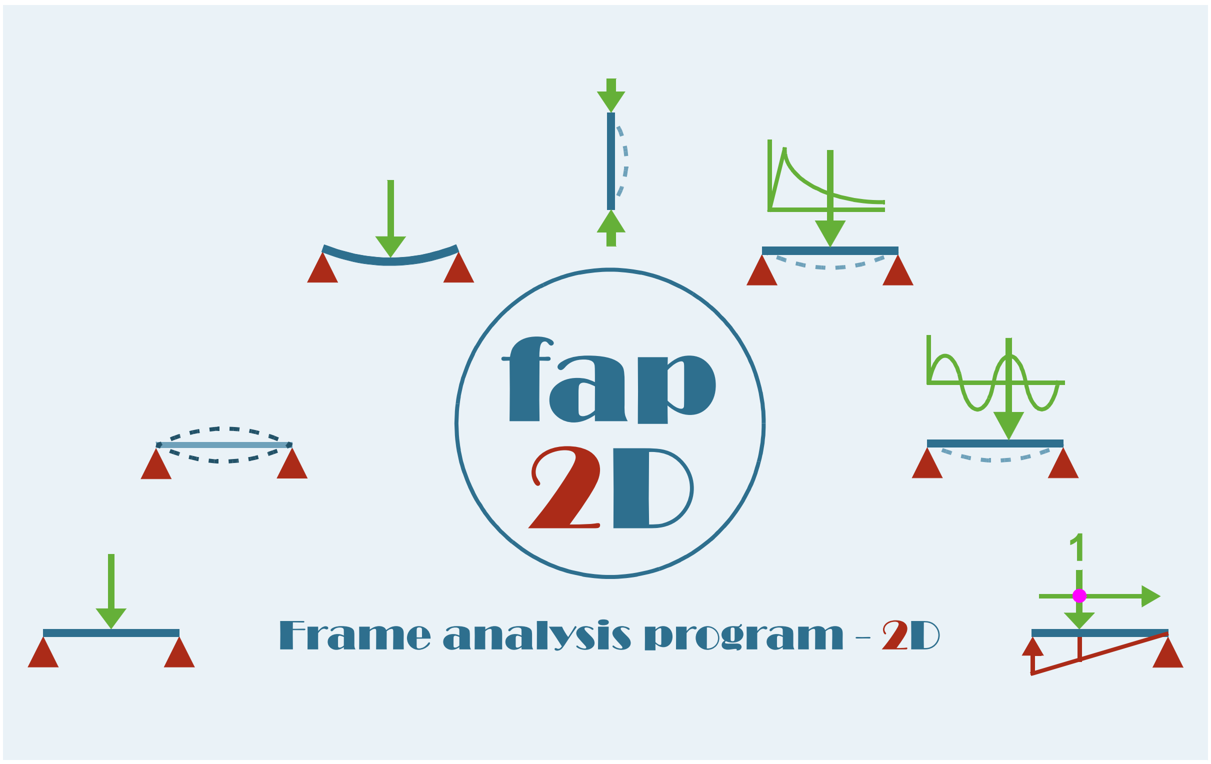


A WINDOWS-based program

for static and dynamic analysis of

2D frame type structures

USER's MANUAL

Preliminary version

Kolbein Bell

May 2011

Content

Preface

The development of program **fap2D** started at the Department of structural engineering at NTNU - Norwegian University of Science and Technology - in 2006, as a combined project/master thesis for two students. The project, which is still ongoing has so far included 8 students under my supervision. These are:

2006/2007 -	Sverre Eide Holst and Magnus Minsaas,
2008/2009 -	Dagfinn Dale Kloven and Gunnar Stenrud Nilsen,
2009/2010 -	Jan Kristian Dolven, and
2010/2011 -	Fredrik Larsen, Brita Årvik and Daniel Aase.

The program consists of two distinct parts, a graphical user interface (GUI) and a "computational engine" (Frame2D). While the computational engine (Fortran code) has been my responsibility, the implementation of the major part of the program, the GUI (C# and OpenGL), has been carried out by the students.

From the start the emphasis has been on the GUI, and our ambition has been to develop a powerful, but above all easy to use analysis tool, suitable for both education and practical engineering work.

I would like to thank all the students who have participated in the project. Your efforts have been impressive, and it has been a pleasure to work with you all.

Trondheim in May 2011

Kolbein Bell
kolbein.bell@ntnu.no

Capabilities

For a qualified user, the short version is that **fap2D** may be used to determine

- the *static response*, according to both linear and nonlinear theory (only geometric nonlinearity is considered),
- the linearized *buckling* load(s) and the associated buckling mode shape(s),
- the *free, undamped vibration* characteristics (frequencies and mode shapes),
- the *dynamic response* due to time dependent loading in the *time domain*,
- the *dynamic response* due to *harmonic loading* in the *frequency domain*,
- the *dynamic response* due to *periodic*, but non-harmonic loading in the *frequency domain*, and
- *influence lines* for specified response parameters due to a "travelling" load on a specified "travel path",

for any valid 2D frame type structure.

This version also includes *design* of *steel* members according to Eurocode 3.

The structure is modelled by straight *beam* or curved *arch members* (circular or parabolic), both of which can accommodate bending moment, shear and axial force, and/or straight *bar*, *cable* (tension only) and *strut* (compression only) *members*, all of which can only accommodate axial force. The members are interconnected at *joints*. Elastic *springs*, both boundary springs and coupling springs, as well as eccentricities, in the form of *rigid*, but weightless "arms" at member ends, may also be included.

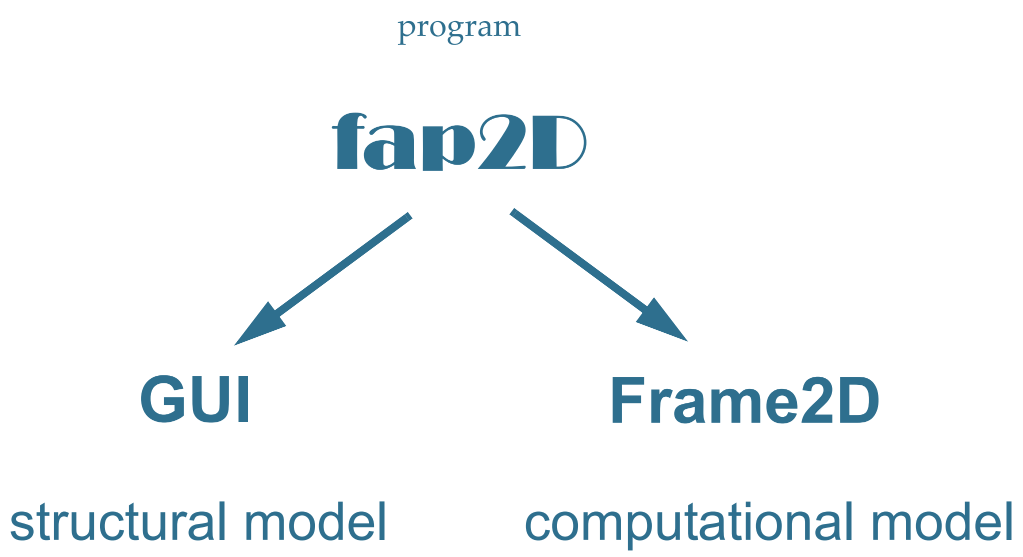
The spatial *loading* may be uniform or linearly varying *distributed load* on beam/arch members, *concentrated loads*, including moments, at joints, *prescribed displacements* at joints and *initial strain* (e.g. temperature). The time variation of spatial loading is defined by several different types of *time functions*.

Boundary conditions, both "external" and "internal" may be specified at joints, in global reference axes or local axes defined at specific joints. An external boundary condition consists of a suppressed degree of freedom (*dof*), whereas an internal boundary condition is a *displacement release* ("hinge") at a joint.

The *computational model*, which consists of only *straight beam elements with constant cross sections*, is generated automatically from the structural model. Each beam and arch member is replaced by 40 (default number) straight Euler-Bernoulli beam elements (no shear deformations) or Timoshenko beam elements (with both bending and shear deformations) with constant cross sections. For short members this number may be somewhat excessive, but it can be easily adjusted, for the entire model or individual members. Bar, cable and strut members are all modelled by *one* beam element (that can only transmit axial force). The user can easily control the number of elements representing a beam/arch member, locally for individual members or globally for all members in the model.
All distributed loading on beam/arch members is *lumped* into statically equivalent concentrated nodal forces.

Computed results are nodal displacements, section forces (*M*, *V* and *N*) for each element, maximum and minimum axial stress at both ends of each element for all types of cross sections and maximum shear stress for most cross sections, as well as residual forces at joints (reaction and "hinge" forces). For some types of analyses, *x*-*y* plot of specified response parameters are available, and for buckling and free vibration analyses buckling factors, vibration frequencies and corresponding modes are determined.

The program consists of two distinct parts, a graphical user interface (GUI) and a "computational engine" (Frame2D).



The GUI is programmed in C# and OpenGL, whereas Frame2D is coded in Fortran, the top level subroutines in Fortran 90 and some lower level (library type) subroutines in Fortran 77.

The user is concerned with the structural model only; she/he can also influence the computational model, but only via the structural model. The user cannot access the computational model directly, except for results.

Units

SI units are used consistently throughout the program.

Platform

fap2d is developed and tested on Microsoft's Windows 7.

It should be emphasized that the current version is a beta version.

The structural model

The structural model, which consists of structural ***members*** interconnected at ***joints***, is referred to a *global* (reference) coordinate system ($\bar{x}, \bar{z}$), see figure 1. The location of the origin of the reference coordinates is defined (arbitrarily) by the user.

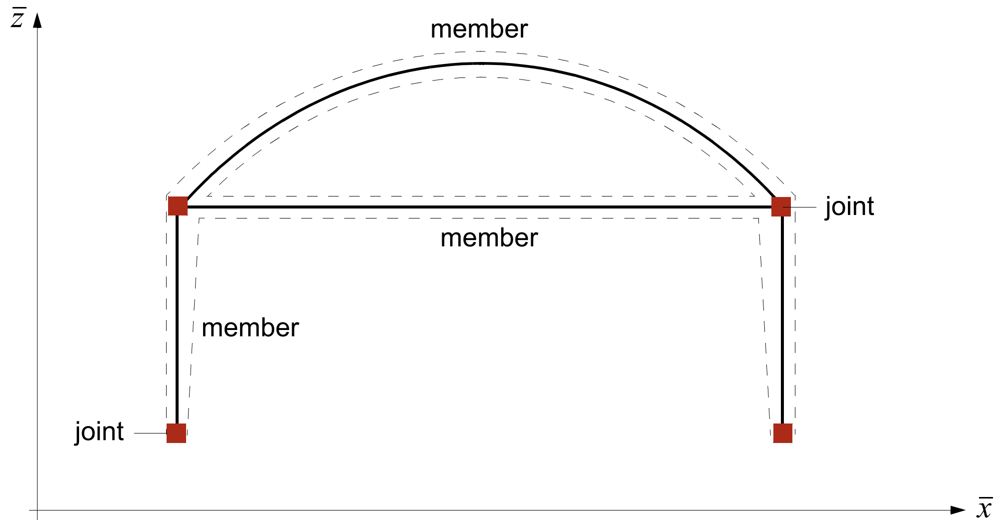


Figure 1 Structural model

A member can also be connected to a point located at the interior of an existing member; such a point is called an ***internal joint*** and its associated member is a ***host member***. Internal joints divides a member into ***sub-members***.

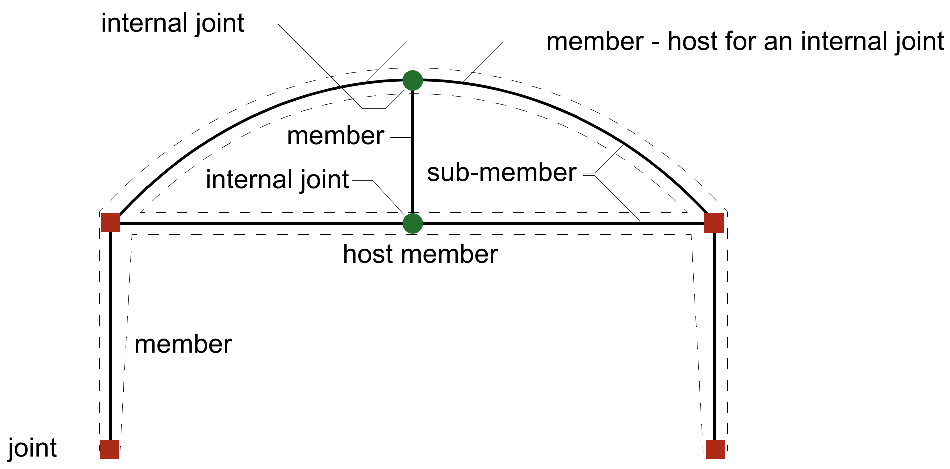


Figure 2 Internal joints and sub-members

Members

The following types of structural members are available:

- ***Straight beam*** members - henceforth called ***beam members***.
- ***Curved beam*** members - henceforth called ***arch members****;*
  arch members may have *circular* or *parabolic* shape.
- ***Straight bar*** members - axial force only - both tension and compression.
- ***Straight cable*** members - axial *tension only* (bi-linear elastic behaviour).
- ***Straight strut*** members - axial *compression only* (bi-linear elastic behaviour).

In addition to these types of members, ***elastic springs***, both *boundary springs* (supporting a particular displacement component) and coupling springs (connecting two similar displacement components or degrees of freedom at coinciding nodes) can also be applied to the structural model.

The ***cross section*** of a member belongs to one of the following three categories:

1. ***Predefined****,* that is standardized (steel) sections whose properties are tabulated.
2. ***Parametric****,* that is a cross section with a geometric form (rectangle, circular tube etc.) that is uniquely defined by a set of "geometric parameters", and for which the necessary "mechanical properties" are determined by the program using the geometric parameters.
3. *General or* ***arbitrary****,* that is, a cross section of arbitrary shape and whose section properties (A, I, etc.) are input to the program (obtained for instance by specialized cross section programs).

For structural members the following rules apply:

a) A particular member, regardless of type, has the same material properties throughout.

b) A particular member, regardless of type, has the same cross section *shape* throughout, and with one exception the cross section is *constant* within the member. The exception is *beam* and *arch* members having cross sections of the *parametric* category. In this case, although the shape is the same, the size can vary from one joint to the next. The individual shape parameters (*e.g.* height, width, radius etc.) of the cross section may vary *linearly* from one joint to the next.

NOTE: An internal joint may define a cross section for its host member, but only if the cross section is of the parametric category, and of the same shape as that of its host member. Hence, it is possible to describe a completely arbitrary variation of the parametric cross sections along beam and arch members.

c) *Straight beam members* can be divided into two or more, shorter, but in all other respects ordinary members that will inherit all properties of the *mother* member, including its loading. Being "ordinary" members means that they can (after "creation") be assigned new properties, irrespective of the mother member.

NOTE: *Arch members cannot be divided into ordinary members.*

NOTE: Both beam and arch members can be divided into *sub-members* by introducing *internal joints*. However, the sub-member remains an integral part of its mother member, and the only properties it can have that differ from those of the mother member are associated with distributed loading.

Figure 3 shows arch members of circular and parabolic type. Regardless of how they are created, they are uniquely defined by the coordinates of the base points A and B, and the radius of curvature (*R*) and height (*h*), respectively. As explained below in the joint section, arch members are somewhat sensitive to changes made to the position of base points A and B and/or *R* and *h*.

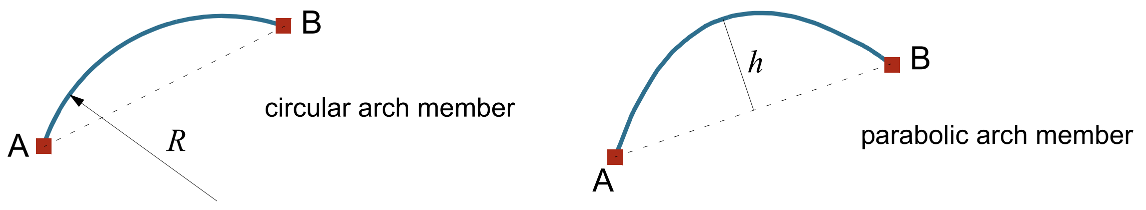


Figure 3 Arch members

Joints

The following rules apply:

1) If a joint is deleted, *all* members connected to it are automatically deleted. However, the opposite does not apply: if *all* members connected to a joint are deleted, the joint is *not* deleted.
It follows from this that joints take precedence over members, *i.e.* members are connected to joints, not the other way around.

2) An *internal joint* resides at the interior of a *host* member. If an internal joint is deleted, *all* members connected to it, except its host member, are deleted.

3) The position of a *joint* may be changed arbitrarily. For an arch member connected to a joint that is moved, the move will have certain consequences explained below. For all other members the move means change of length and/or orientation.

4) The position of an *internal joint* may be changed, but only *along* the host member axis.

5) *Internal joints* may be introduced in one of four ways:
a) by placing it on the axis of an existing beam or arch member, which then becomes a host member for the internal joint,
b) by subdividing an ordinary beam or arch member,
c) by joining a new member to a point located at the interior of an existing member; the latter becomes the host member for the internal joint, and finally
d) by applying a concentrated load to a point "inside" a member (concentrated or point loads can only act at a joint or internal joint).

NOTE: An internal joint can only have one host member.

What happens if the shape of an arch is changed, by moving one or both of its base points (A and B) and/or by changing the parameter R or h?

If the member has no internal joints, there is no problem. The member's geometric definition is simply updated. If, however, the member has internal joints, a well defined and unique procedure for the new location of the internal joints is necessary. By definition, they reside on the new position of the member axis, and their *relative distance* from the first base point (A) is the *same* as before the change, measured along the chord (which is the straight line between A and B).

Both members and joints are numbered, in the order they are created. While internal joints are included in the joint numbering series, the sub-members are not numbered. The numbers may be shown or hidden. Depending on how the model is established, the numbering can be quite erratic and it may thus prove useful to use the renumbering facility (located in the toolbox).

Boundary conditions

Boundary conditions are defined at joints. All joints and internal joints will become nodal points in the computational model and as such each has three kinematic degrees of freedom, two orthogonal displacements and one rotation. Any one degree of freedom may be

- *free* or unknown,
- *suppressed*, which means it has a fixed value of zero,
- *prescribed*, which means it has a fixed, non-zero value, or it may be
- *dependant* of (coupled to) another (free) degree of freedom.

Figure 4 shows the various possibilities of suppressing degrees of freedom at a joint, along with their symbols.

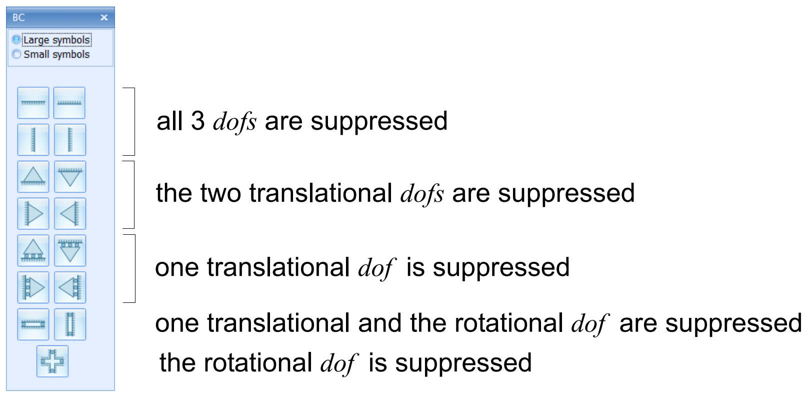

BC
Large symbols
Small symbols
all 3 *dofs* are suppressed
the two translational *dofs* are suppressed
one translational *dof* is suppressed
one translational and the rotational *dof* are suppressed
the rotational *dof* is suppressed

Figure 4 Suppressed degrees of freedom

A prescribed *dof* can only be imposed on a suppressed *dof*. In other words, a prescribed *dof*, which in most cases will result in an indirect loading effect, is imposed by "moving" a support.

A dependant *dof* is defined by the simple constraint equation:

$$r_s = r_m$$

where the dependent or *slave dof*, r_s , is set equal to a *master dof*, r_m , which *must* be a *free dof*. This simple slave concept is the tool provided for modelling all types of "hinges" or displacement releases. However, the details of this are all well hidden for the user. The constraint equations required to handle a displacement *release* are automatically created by the program once the user has defined (in a fairly intuitive way) the kind of release he or she wishes to impose.

It should be noted that the degrees of freedom at a joint follow the coordinate axes at the joint. This will be the global axes unless the user has defined local axes at the joint, which she/he can do at any joint of the model.

Suppressed, prescribed and dependent *dofs* are collectively referred to as *specified dofs*. By this terminology we have two types of *dofs*, *free* (or unknown) *dofs* and *specified* (or known) *dofs*.

Instead of, or in combination with, specified *dofs*, elastic springs may be used to simulate boundary constraints. One example of effective combination is the modelling of a *semi-rigid* joint. A *coupling spring* may be attached to the degrees of freedom created at a joint through a *releasing* "hinge".

Eccentricities

Eccentricities and very stiff parts of a structure are most conveniently (and safely) modelled as completely *rigid links* or arms at the end of one or more members. A stiff corner of some size in a frame, as indicated by figure 5a, may, for instance, be modelled as shown in figure 5b. The red "arms" are completely rigid and weightless links. Such links may be introduced at the end of any member in the model, and they need not be mere extensions of the member. In other words, the rigid links may have directions that are completely independent of the member they are attached to.

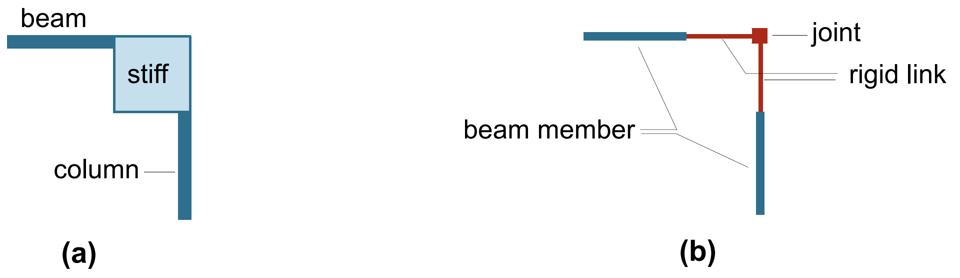


Figure 5 A stiff frame corner (**a**) and its model (**b**)

In many cases it is better to model something very stiff as completely stiff, using rigid links, instead of one or more very stiff members.

Spatial loading

The following types of *spatial* loading may be applied:

1. *Distributed loading*
   Four types of distributed loading:
   - *gravity* loading
   - *projection* loading (both horizontal and vertical)
   - *normal* loading

- *tangential* loading
may be applied to any member, see figure 6.
All distributed loads can have a *linear variation* along the member or its projections in the horizontal or vertical direction.
Furthermore, a distributed load may be applied to the entire member or to one or more sub-members.

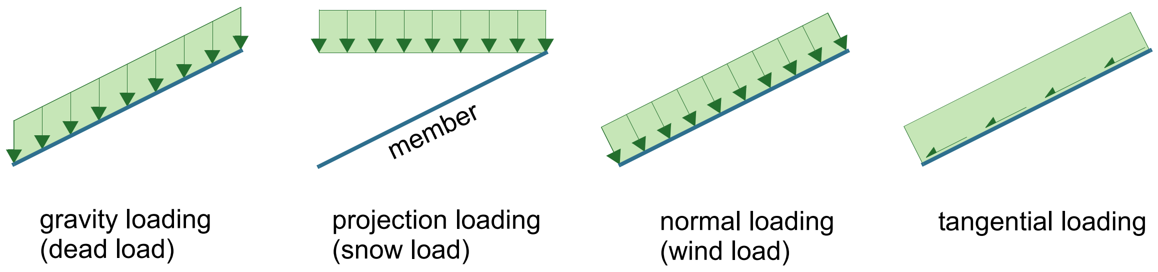


Figure 6 Distributed loading

2. *Concentrated loads (including moments)*
Concentrated (point) loads, in the direction of the global reference axes, can be applied anywhere within beam and arch members, but only at member ends for bar, cable and strut members. Concentrated moments can only be applied to beam and arch members.
Concentrated (point) loads can only be applied at joints or internal joints. If a point load is applied "inside" a beam or arch member where there is no internal joint, such a joint will be automatically inserted by the program.

NOTE: All external loading is *conservative*.

3. *Initial strain* (*e.g.* temperature) can be applied to any member. For each member an initial strain condition that is constant in the element direction, but may have a linear variation over the member height, may be specified, see figure 7.
4. *Prescribed displacements* also have a load effect. However, prescribed displacements are treated as boundary conditions and as such are dealt with in the following section.

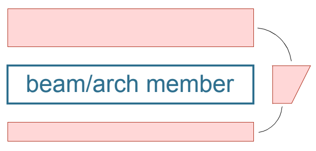


Figure 7 Initial strain

Each external load, whether distributed or concentrated, must be assigned to a *named* ***load case*** (**LC**). The user may define any number of distinctly named load cases, and any one load case can contain any number of individual loads. Two **LC**'s are created automatically, one called Default load case and one called Own weight. The first will accommodate all external loading not specifically assigned to another (named) **LC**, whereas the latter will accommodate the own weight of the structural model, a loading that is automatically computed by the program. If one or more prescribed displacements are specified by the user, these will be associated with an automatically created **LC** called Prescribed displ. Similarly, if temperature and/or any other form of initial strain is defined, *all* such loading is accommodated by an automatically created load case called Init. strain. It should be noted that a specific model can only have one **LC** (Prescribed displ.) for prescribed displacements and one **LC** for initial strain/temperature (Init. strain).

Computations are carried out for named ***load combinations*** (**LCmb**), not load cases. A spatial load combination is a linear combination of any number of named **LC**'s. Each selected **LC** contributes by a user specified constant *load factor*.
The user may define any number of distinctly named load combinations, and any one load combination can contain any number of individual **LC**'s. The program creates automatically an **LCmb** called Default load combination, which, on creation, contains only the Default load case with a load factor of 1,0. If no loads have been assigned to the Default load case, the Default load combination consists of the load case Own weight times 1,0. It follows from this that it will always be possible to carry out a (linear) static analysis (for own weight only) without having specifically defined any loading for the model.

NOTE: If prescribed displacements have been specified (for suppressed *dofs*) the Prescribed displ. **LC** has been created. An analysis carried out for a load combination that do *not* include this **LC**, will assume that the *dof(s)* in question is (are) suppressed.

Time dependent loading

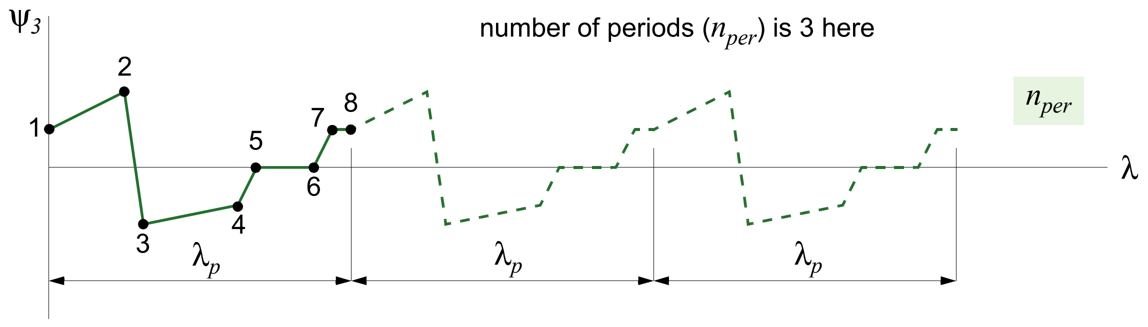
Time, defined by the time parameter λ, means real time (in seconds) if we are talking about a dynamic analysis, whereas it is pseudo-time in the case of a nonlinear static analysis (used to define the loading and response history).

Time dependent loading consists of a spatial load combination multiplied by a time-dependent load factor ψ which is referred to as a *time function*. The program recognizes 5 different *time function types*, valid between 0 and λ_{max}:

Type 1: $\psi_1 = 1{,}0$ (constant)

Type 2: $\psi_2 = \dfrac{\lambda}{\lambda_{max}}$ (linear between 0 and 1)

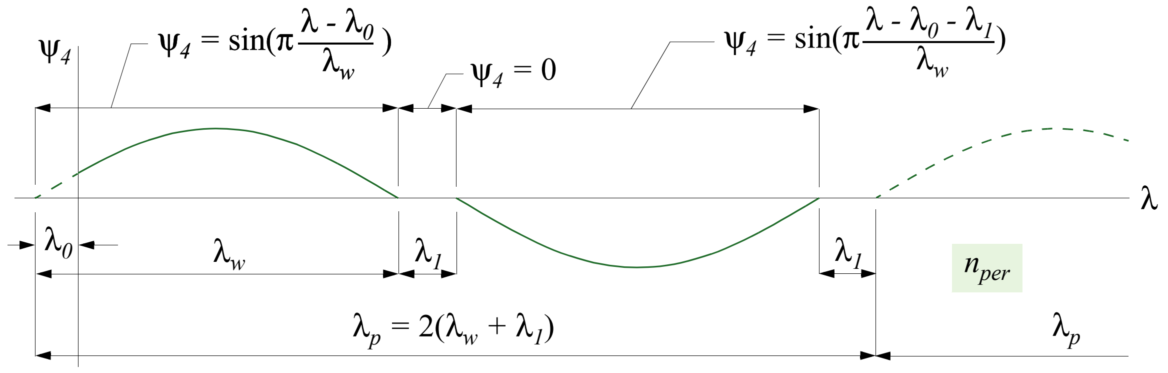
Type 3: Arbitrary, but possibly semi-periodic:



ψ_3 is defined by n_p pairs (λ_i, ψ_{3i}) of numbers and the number (n_{per}) of periods, all of which are input information. The λ-values are measured in terms of time units. The function value varies linearly between the points and the requirements are:

$$\lambda_{i+1} > \lambda_i \quad \text{and} \quad (\lambda_p \times n_{per}) < \lambda_{max}$$

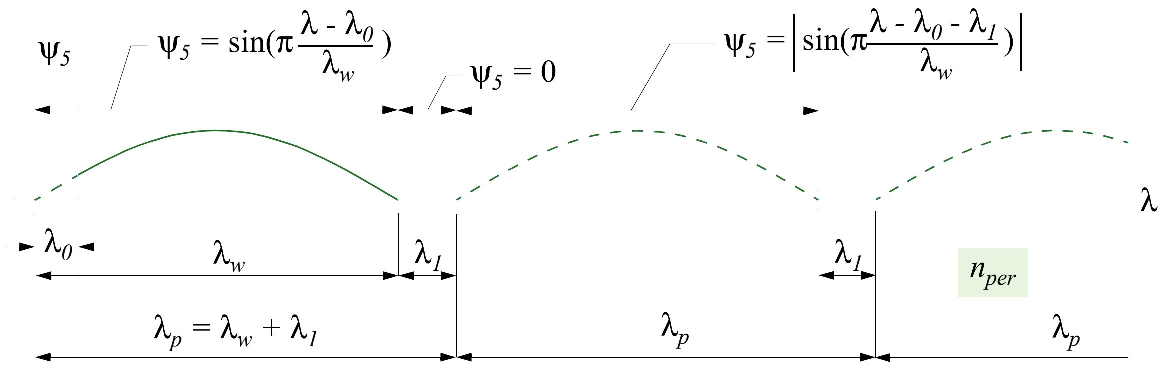
Type 4: Sine function A:



ψ_4
$\psi_4 = \sin(\pi \frac{\lambda - \lambda_0}{\lambda_w})$
$\psi_4 = 0$
$\psi_4 = \sin(\pi \frac{\lambda - \lambda_0 - \lambda_1}{\lambda_w})$
λ
λ_0
λ_w
λ_1
λ_1
n_{per}
$\lambda_p = 2(\lambda_w + \lambda_1)$
λ_p

In the figure above λ_0 is a negative number. If $\lambda_0 > 0$ then $\psi_4 = 0$ for $\lambda \leq \lambda_0$. All λ-values are measured in terms of time units.

Type 5: Sine function B:

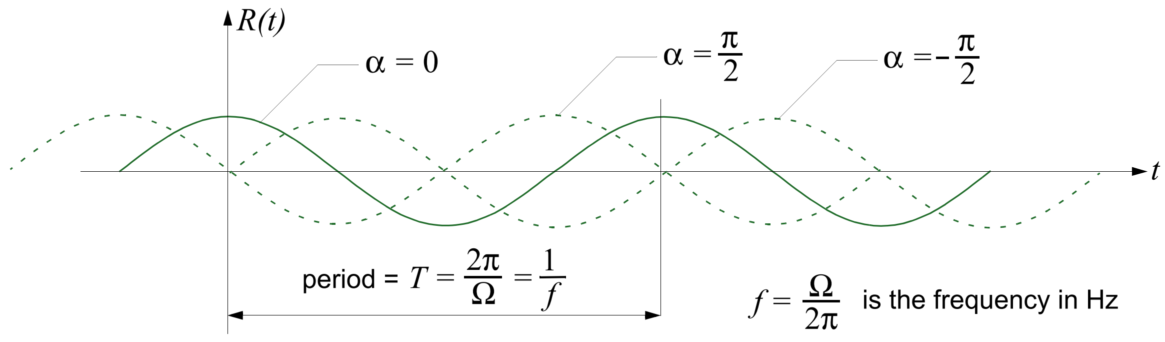


ψ_5
$\psi_5 = \sin(\pi \frac{\lambda - \lambda_0}{\lambda_w})$
$\psi_5 = 0$
$\psi_5 = \left| \sin(\pi \frac{\lambda - \lambda_0 - \lambda_1}{\lambda_w}) \right|$
λ
λ_0
λ_w
λ_1
λ_1
n_{per}
$\lambda_p = \lambda_w + \lambda_1$
λ_p
λ_p

Types 1 and 2 are uniquely defined by their type numbers (they require no other information than λ_{max}).

Type 3 requires n_p pairs of numbers (λ_i and ψ_{3i}) plus the number of periods n_{per}, whereas types 4 and 5 require λ_0, λ_1, λ_w and n_{per}.

Frequency dependent loading



$R(t)$
$\alpha = 0$
$\alpha = \frac{\pi}{2}$
$\alpha = -\frac{\pi}{2}$
t
period = $T = \frac{2\pi}{\Omega} = \frac{1}{f}$
$f = \frac{\Omega}{2\pi}$ is the frequency in Hz

Figure 8 Harmonic loading

Frequency dependent loading is *harmonic*, with a variation along the time axis as shown in figure 8. The load frequency is Ω (rad/s) or *f* (Hz). **fap2D** recognizes two types of harmonic loading:

1. Only *one* load frequency (Ω). In other words, all loading is harmonic in time and have the same frequency. However, more than one spatial load combination may be applied, and these may have different *phase angles* (α). In fact, if the phase angles are not different, it makes little sense to apply more than one spatial load combination, and all spatial loading may as well be lumped into one load combination.
2. Only *one* spatial load combination is applied as a harmonic load, but it may be applied with many different frequencies, from Ω_1 to Ω_{max}, in equal steps of ΔΩ. In this case phase angle has no meaning for the loading.

Periodic, but non-harmonic loading

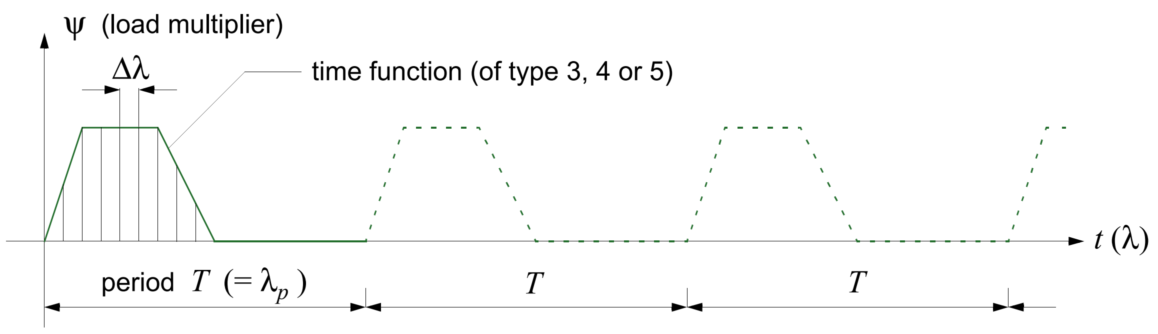


Figure 9 Periodic loading

This loading is characterized by one or more spatial load combinations and a *time function* ψ, the *same* time function for *all* contributing loading (which can be due to external loading and prescribed displacements). The time function must be of type 3, 4 or 5, see above section about time dependent loading.

The periodic loading, exemplified in figure 9, is automatically replaced by a series of *harmonic load combinations* through a Fourier series analysis. The number of Fourier terms included is determined by a user defined tolerance parameter, subject to a “ceiling” defined by the maximum permissible number of terms, which is also specified by the user.

Mass

The mass of the structural members may be (automatically) accounted for by one of three mass representation models:

- *lumped* mass representation - the mass of each element is lumped into two equal concentrated “translational masses” at the element nodes; all rotational *dofs* are mass-less,
- *consistent* mass representation - the element mass matrix is established on the basis of the same displacement functions as the stiffness is derived from; this leads to rotational as well as translational mass, and mass coupling,
- *diagonalized* mass representation - this is a combination of the other two models; it leads to a diagonal element mass matrix, *i.e.* both translational and

rotational mass, but no mass coupling.
NOTE: The theoretical basis for this model is not very well founded.

The basic modelling philosophy adopted by the program (see next chapter) favours the lumped mass approach, and only numerical reasons seem to warrant one of the other two methods. In some circumstances the lumped approach may lead to numerical difficulties (in the solution of the free vibration eigenproblem).

In addition to the mass of the structural members, concentrated (translational and rotational) mass may be introduced at joints, including internal joints.

Damping

Dynamic analysis may be carried out on the complete (coupled MDOF) model or on a reduced (decoupled SDOF) model obtained through use of a limited number of *modal coordinates*. Regardless of model, *viscous damping* is assumed.

For the complete MDOF model the available damping model is the so-called *Rayleigh damping* in which the damping matrix **C** is expressed as a combination of the mass matrix (**M**) and the stiffness matrix (**K**) of the model, that is

$$\mathbf{C} = a_1\mathbf{M} + a_2\mathbf{K}$$

The coefficients a_1 and a_2 may be given explicitly (as input) or they may be computed by the program on the basis of generalized mass and stiffness; more about this later.
The user can specify mass proportional damping ($a_2 = 0$), stiffness proportional damping ($a_1 = 0$) or a complete Rayleigh damping (both a_1 and a_2 have non-zero values).
For a complete MDOF model it is also possible to include "point dampers" (viscous dashpots), at any (free, non-specified) *dof* of any joint, in addition to or instead of the Rayleigh damping.

For an SDOF model (that is modal analysis), damping ratios may be specified explicitly (as input) for each contributing mode, or alternatively the damping ratios may be computed implicitly for each mode using a Rayleigh type approach; more about this later. For this (SDOF model), point dampers *cannot* be included.

The computational model

Basic philosophy

Once the structural model and its spatial loading is complete one of several analyses may be specified. Depending on the type of analysis some more information may be needed (mostly concerning the loading) before the analysis can be started; more about this later. As and when an analysis starts the structural model is (automatically) converted into a computational model. This transformation is based on the philosophy indicated by figure 10. Beam and arch members are subdivided into a fairly large

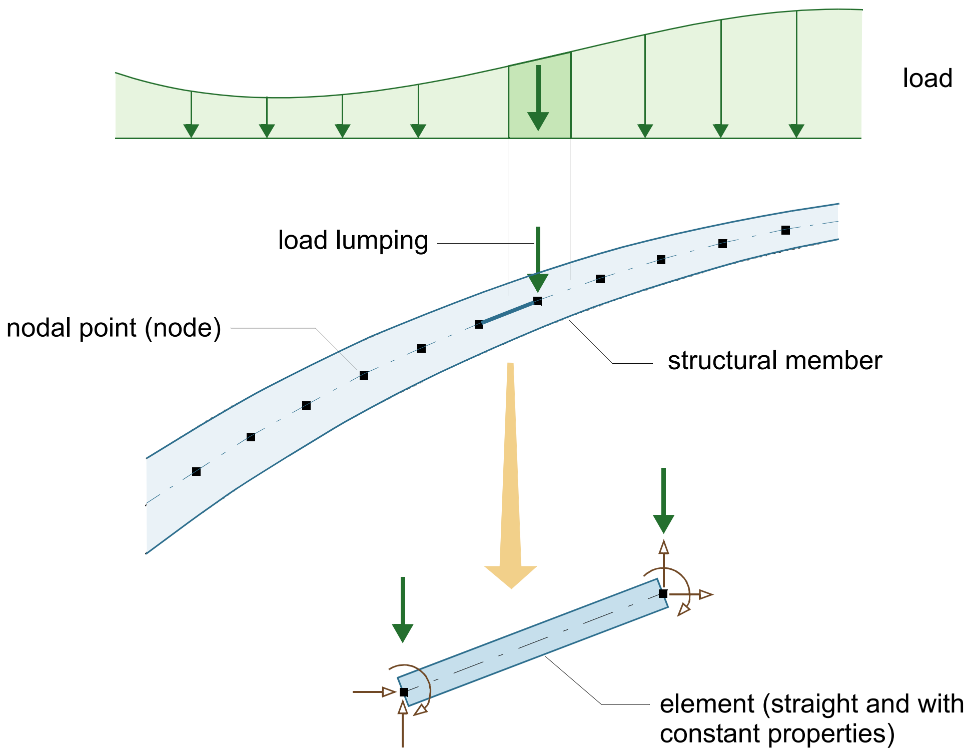


Figure 10 Basic modelling concept

number of straight *beam elements*, each with 6 degrees of freedom and constant cross section properties (determined as the properties at the element's mid-point). Distributed loading, if present, is lumped into statically equivalent concentrated loads at the nodes.

For the structural member in figure 10, the number of elements required is probably dictated by the member geometry. However, the idea of load lumping also requires a straight beam member to be subdivided into a series of shorter elements if it is subjected to any form of distributed loading (even if the geometry does not call for such subdivision). For a straight member the number of computational elements is dictated by the load representation and possibly also by a varying cross section (which is approximated by step-wise constant section properties).

This simple strategy leads to a much higher number of degrees of freedom, and thus more numerical work and higher storage demands, than the more conventional approach of one to one relation between member and element. The simplicity of the "brute force" technique, combined with some obvious advantages in describing curved members and geometric imperfections, is believed to more than compensate for the increased computational effort and storage space. It also lends itself extremely well for geometric presentation of both model and results - everything boils down to simple straight lines.

Reference and identification

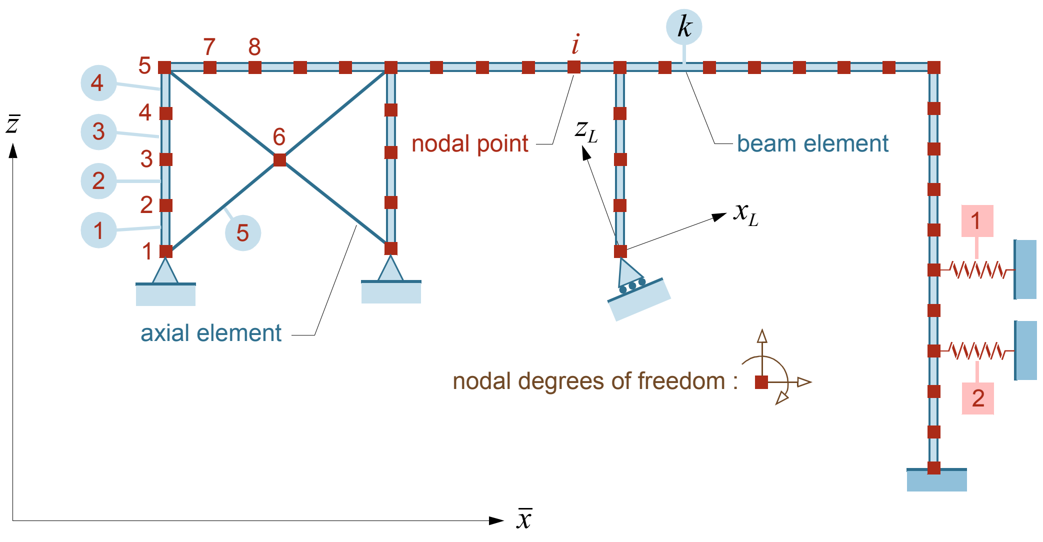


Figure 11 Typical computational model

With reference to figure 11, the *computational model* consists of straight beam and axial *elements* interconnected at *nodal points* (or just *nodes*). Elastic *spring* elements may also be included. The model is referred to a *global* or reference coordinate system $\bar{x}, \bar{z}$. Each nodal point is assigned a unique number, ranging from 1 to number of nodes, and each element is numbered consecutively from 1 to number of elements. This latter number series includes both beam and axial (bar, cable, strut) elements, in any order, but spring elements are numbered in a separate series. This numbering does not really concern the user; furthermore the node numbering is automatically "optimized" with respect to equation solving by a fairly efficient renumbering scheme.

Each nodal point has three kinematic degrees of freedom (*dofs*), two orthogonal translations and one rotation. By default the *dofs*, also denoted nodal displacements, are referred to (are parallel with) the global reference axes, $\bar{x}$ and $\bar{z}$. It is, however, possible to define a *local* coordinate system, x_L, z_L, at any node. At such a node the translational *dofs* follow the local axes.

At a node where only axial elements meet, *e.g.* node 6 in figure 11, the program automatically suppresses the rotational *dof* (which does not receive stiffness contributions from any of the axial elements).

Elements

The following elements are available:

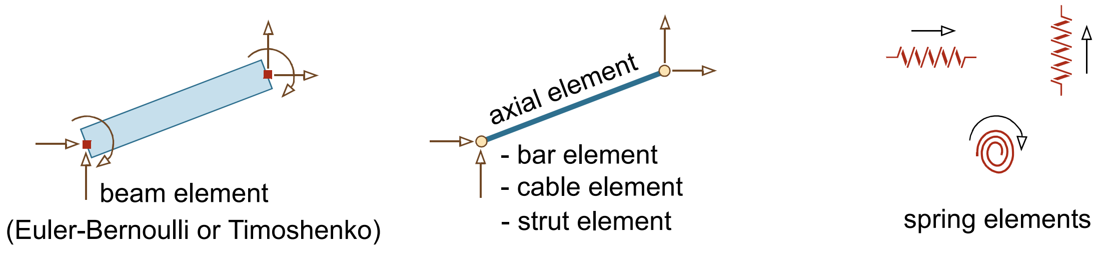


Figure 12 Elements in **fap2d**

Beam element - a simple, straight 6-degree-of-freedom element with constant cross section. Bending, axial and shear deformations are considered. The latter, which is based on the assumption of an average shear deformation (Timoshenko theory), is optional.
Material properties are *linearly elastic*, for all types of analysis.

Axial element - a simple, straight 4-degree-of-freedom element with constant cross section that can only take axial force. *Bi-linear* stiffness characteristics may be specified. In other words, a particular axial element may take both tension and compression, in which case it is a *bar* element, tension only, in which case it is a *cable element*, or compression only, in which case it is a *strut element*.

Spring elements - both linear and rotational springs may be included in the model. A spring may be a
- *boundary spring* which is a spring connected to a single degree of freedom at one end and fixed ("to earth") at the other, or a
- *coupling spring* which is a spring connecting the same *dof* at two nodal points, *e.g.* the rotation at two nodes - it should be noted that the two nodes coincide (geometrically).

For most types of analyses the springs are linear, but nonlinear springs are also available (for nonlinear static analysis), see figure 13. In *all* cases the springs have *identical* characteristic in tension and compression.

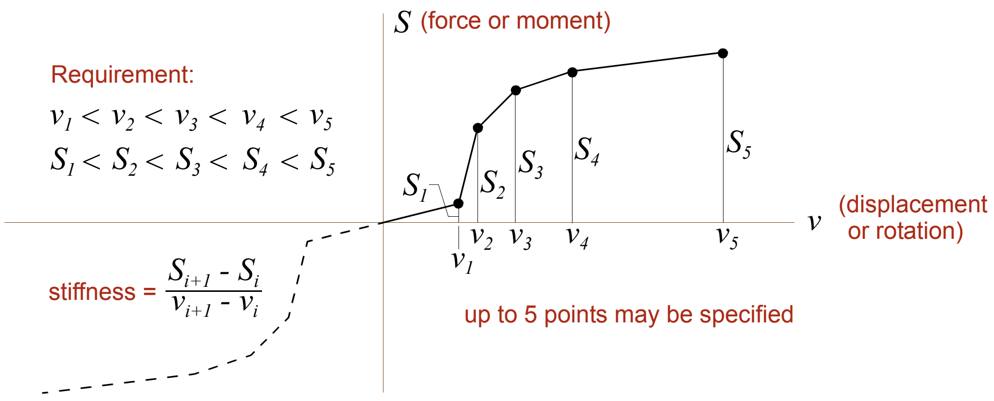

S (force or moment)
Requirement:
$v_1 < v_2 < v_3 < v_4 < v_5$
$S_1 < S_2 < S_3 < S_4 < S_5$
S_1
S_2
S_3
S_4
S_5
v_1
v_2
v_3
v_4
v_5
v
(displacement or rotation)
stiffness = $\frac{S_{i+1} - S_i}{v_{i+1} - v_i}$
up to 5 points may be specified

Figure 13 Nonlinear spring stiffness

Solution

The system matrices, stiffness (**K**), mass (**M**) and load (**R**), are assembled to include only the unknown degrees of freedom; in other words, all specified *dofs* are omitted from the matrices. Stiffness matrices and consistent mass matrices are stored in so-called *skyline* storage format, and the basic numerical operation of solving a system of linear algebraic equations is accomplished by direct GAUSSIAN elimination ($\mathbf{LDL}^T$), through factorization and substitutions.

For the *eigenvalue* problems (free vibration and linearized buckling) the user can choose between *subspace iteration* (which is the default method) and a truncated algorithm due to LANCZOS. The latter is by far the most efficient with regard to computational effort; however, subspace iteration is a well tested and fairly robust algorithm.

More computational details are given below for the individual types of analysis.

Modelling structure and loading

GUI basics

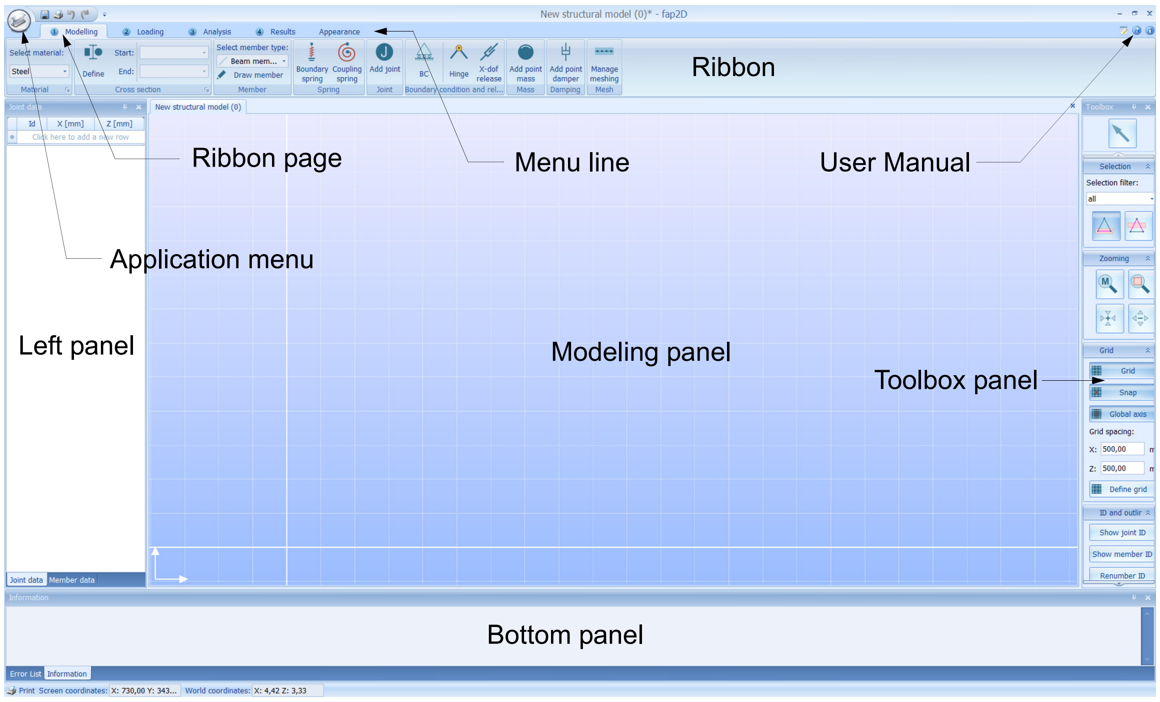


Figure 14 GUI overview

Figure 14 shows an overview of the GUI. The panels shown in the figure are also referred to as “dock panels” since they may be docked anywhere in the GUI view. The first thing a new user has to do once the program provides the **fap2D** window is to push the application menu button at the top left corner. This will launch the menu shown in figure 15. Recent structural models will, at this stage, be empty and the natural choice is therefore to push “New” which produce the window shown in figure 14.

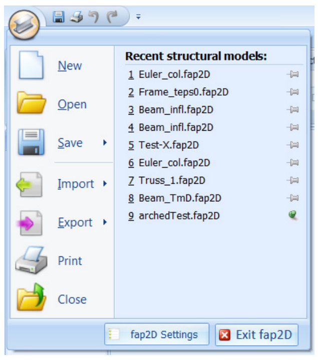


Figure 15 Application menu

The program makes use of the *ribbon* concept, and basically, the user works from left to right. Apart from this manual, a pdf-version of which is available from the question mark button at the right-hand top corner, there is not much in terms of “help” available. The main design criterion has been to make the use of the program as intuitive as possible through familiar icons and well designed dialog boxes. Tooltips are provided where deemed necessary/useful.

On the whole the left-hand mouse button is the "operation" button (apology to all left-handers!) and the right-hand button is the "information" button.

The menu line has four main choices, `modelling`, `loading`, `analysis` and `results`. The fifth choice (`Appearance`) has to do with the style and coloring of the views.

Modelling the structure

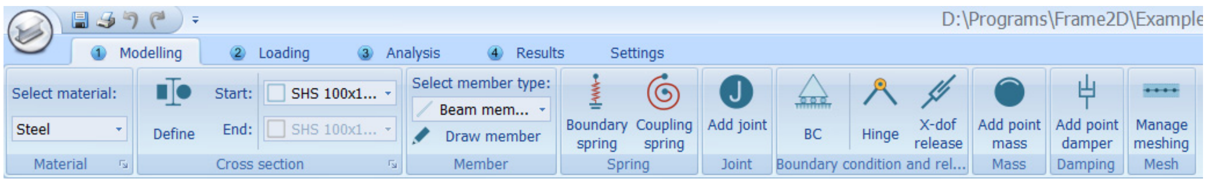


Figure 16 The modelling ribbon

How to establish a viable structural model is fairly straightforward; again the "natural" mode of operation is from left to right, see the modelling ribbon in figure 16. The program has predefined 4 materials: Steel, Concrete, Timber and Aluminum, all with typical parameters. However, the user may define hers or his own material types by selecting Add/edit in the pull-down menu. Next all cross sections to be used in the model should be selected (if of predefined category) or defined (if of parametric or arbitrary category).

When placing a joint, explicitly via the Add joint button or implicitly as the start or end point of a member, it should be kept in mind that the program default is to snap the joint to the closest grid point. Grid spacing and snap can be controlled from the toolbox, but it is also quite straightforward to change the coordinates of a joint once it has been created. It should also be noted that once a specific function has been chosen (for instance by pushing a button), this function remains active ("in the pointer") until a new function or the neutral pointer is chosen. Figure 17 shows the structural model of a simple frame. The "hinge" at point D "decouples" the rotation of the column

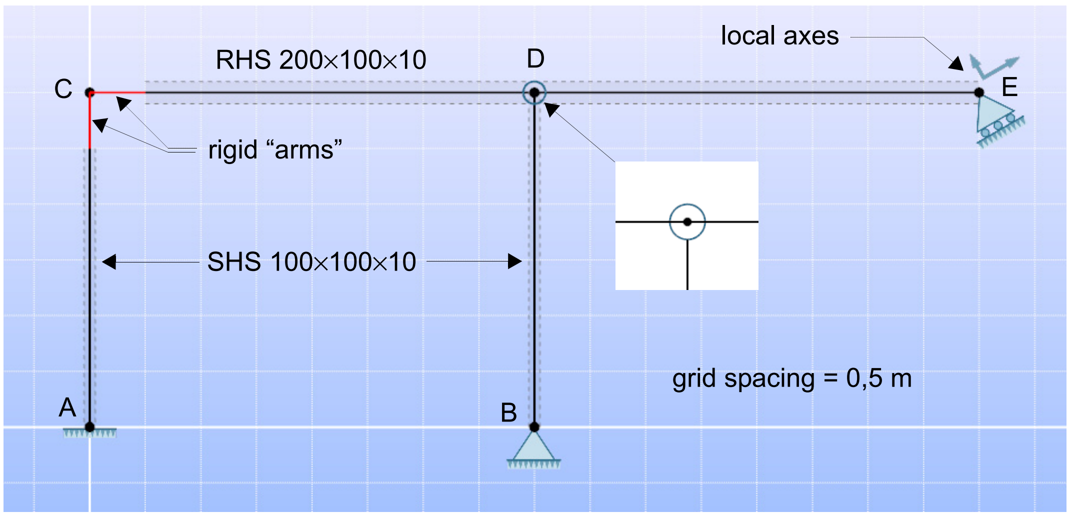


Figure 17 Structural model of a simple frame

from that of the beam which is continuous over the column.

In order to include *local coordinate axes* (point E), *eccentricities* (rigid links at point C) or *response parameters*, right-click the joint and select the appropriate function from the popup menu which in turn will lead you to a fairly self-explanatory dialog box.

Modelling the loading

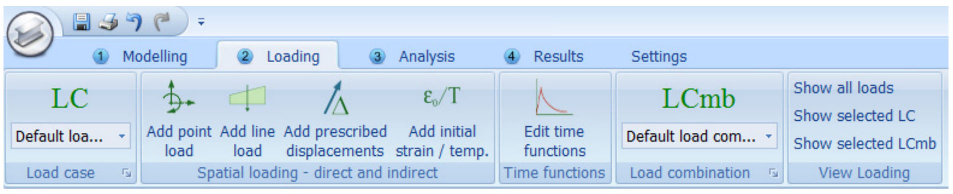
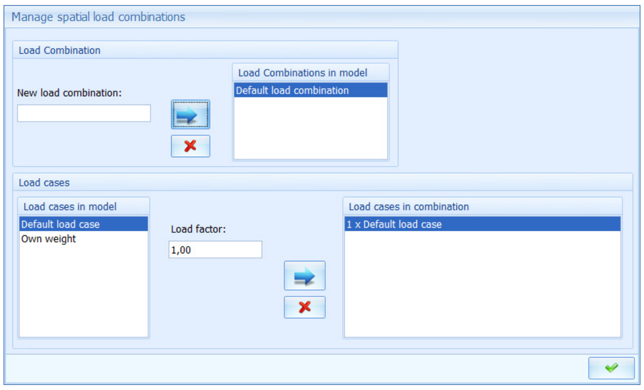


Figure 18 The loading ribbon

Also the loading ribbon, shown in figure 18, is fairly self explanatory; *spatial* loading, in terms of *load cases* (**LC**), and *time functions* are defined here. It should be noted that prescribed displacements and initial strain both gives rise to spatial loading. All spatial loading must be assigned to a particular **LC**. Two predefined load cases exist, Default load case and Own weight, and the user may define any number of named load cases via the Add/edit command in the drop down menu. Own weight contains the dead load of the structural members themselves, and this loading is automatically computed by the program; other loading cannot be assigned to this **LC**. The Default load case and any user defined **LC** can accommodate any number of concentrated loads/moments and/or distributed member loads.

If prescribed displacements are defined (at one or more supported joints) the program automatically creates an **LC** named Prescribed displ.; *all* prescribed displacements of a particular structural model will be associated with this **LC** which can only hold prescribed displacements (no other loading). Similarly, if temperature or other types of initial strain is defined for one or more members, the program automatically creates an **LC** named Init. strain; *all* loading of type initial strain will be associated with this **LC** which cannot hold any other type of loading.

Named time functions of types 3, 4 or 5 (see the section on time dependent loading on page 13) are defined via the Edit time functions button.


Manage spatial load combinations

Load Combination

New load combination:

Load Combinations in model
Default load combination

Load cases

Load cases in model
Default load case
Own weight

Load factor: 1,00

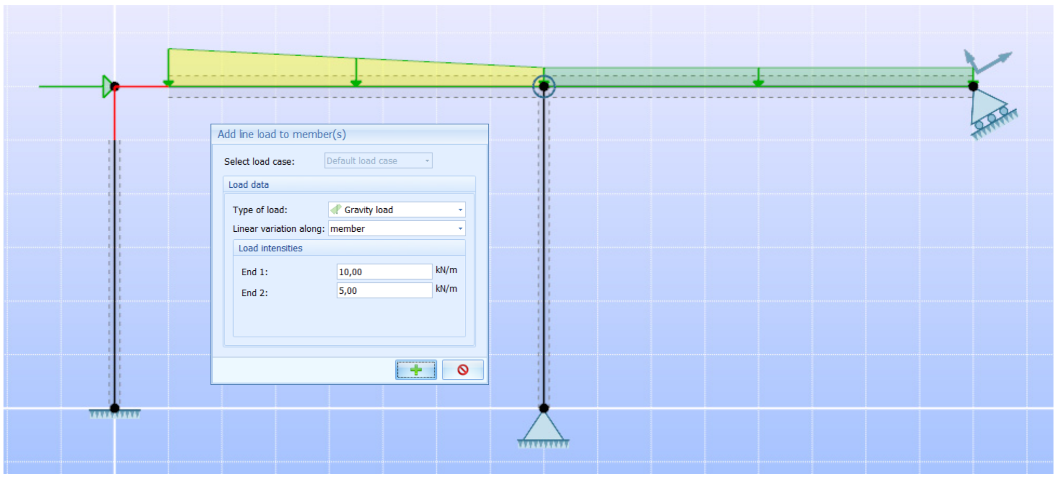
Load cases in combination
1 x Default load case

Figure 19 Dialog box for creating and defining spatial load combinations

Spatial load combinations (**LCmb**) may be defined via the LCmb button in the Loading ribbon, see figure 19. The program has one predefined load combination, Default load combination, which initially contains 1,0×Default load case, if Default load case contains at least one load; if not, Default load combination contains 1,0×Own weight

It should be noted that while spatial load combinations can also be defined in the *Analysis* ribbon (for some analysis), spatial load cases and time functions can only be defined here in the *Loading* ribbon.

Figure 20 shows a spatial LC for the frame of figure 17; for convenience we have assigned the loading to the Default load case. It should be noted that rigid arms or links *cannot* take distributed loading; any such loading must be transferred, by the user, to the joint at the end of the link (as one or two forces, depending on the link's orientation, and a moment).


Add line load to member(s)

Select load case: Default load case

Load data

Type of load: Gravity load

Linear variation along: member

Load intensities

End 1: 10,00 kN/m

End 2: 5,00 kN/m

Figure 20 A spatial **LC** (Default load case)

Analysis and results

Linear static analysis

Figure 21 shows the ribbon for linear static analysis. The Run analysis arrow is active, and if pressed the analysis will be carried out without shear deformations for the Default load combination. Shear deformations are either neglected (which is default) or included by pressing the Include shear deformation button; this button toggles off/on. The LCmb drop down menu enables the user to select any existing load combination for the analysis; in fact the user can also define new load combinations or edit existing ones from this position (the last item in the drop down menu is Add/edit), but only using existing load cases (**LC**). If new **LC**s are required it is necessary to go back to the *Loading* ribbon.

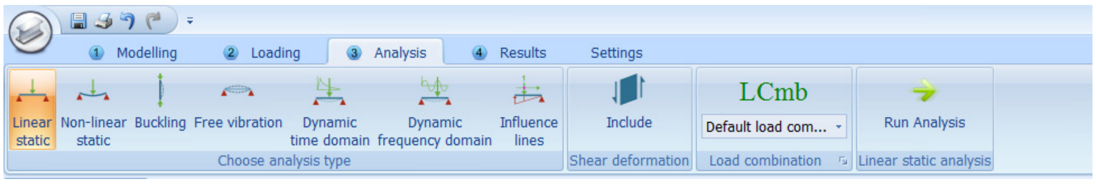


Figure 21 Analysis ribbon for linear static analysis

Computational aspects

The computations are straightforward, and the analysis is carried out for one load combination at a time. By default, shear deformations are *not* included, but as already mentioned above they are easily included by simply pushing the Include shear deformation button before the Run Analysis button. If *bilinear* axial members, that is *cable* and/or *strut* members, are present, the model is not strictly linear. In this case the program will make sure, through an iteration procedure, that all cable and strut members carry tension or compression, respectively. During the iteration procedure such members may thus be “removed” from or inserted back into the model, and the iteration continues until no bilinear members needs to be removed/inserted from one iteration to the next.

Typical results

The frame of figure 17 subjected to the loading shown in figure 20 is analysed. The results available are shown in figure 22. The main results are diagrams of displace-

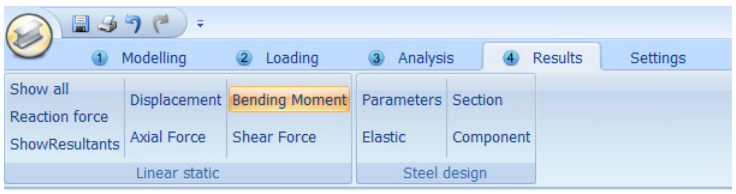
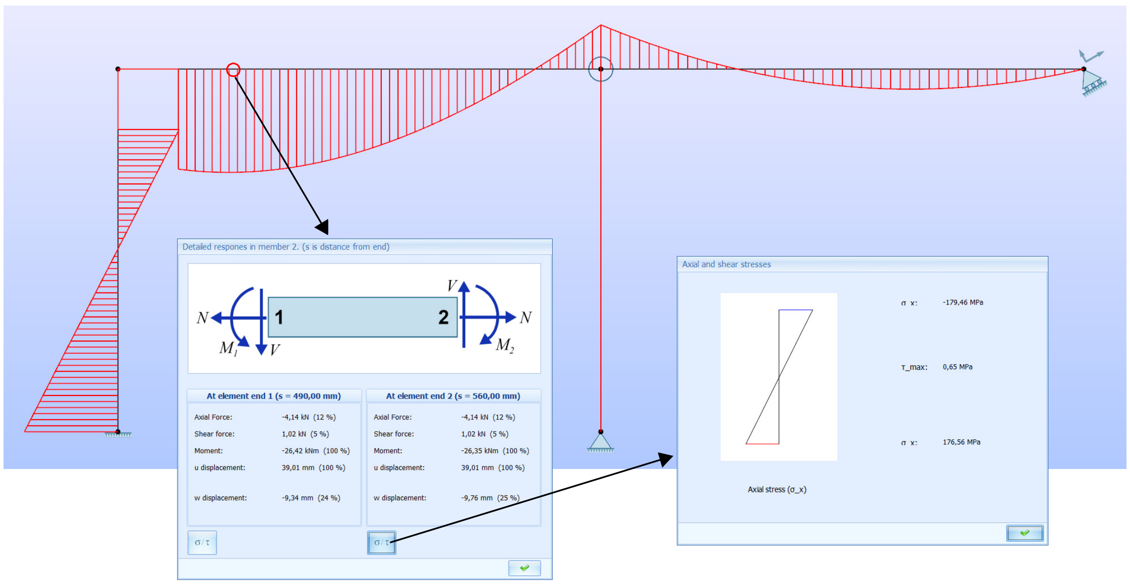


Figure 22 Results available for a linear static analysis

ments and section forces (N, M and V). These can be shown, all four at once (Show all) or one diagram at a time. Figure 23 shows the bending moment diagram. It



Detailed respones in member 2. (s is distance from end)

At element end 1 (s = 490,00 mm)		At element end 2 (s = 560,00 mm)	
Axial Force:	-4,14 kN (12 %)	Axial Force:	-4,14 kN (12 %)
Shear force:	1,02 kN (5 %)	Shear force:	1,02 kN (5 %)
Moment:	-26,42 kNm (100 %)	Moment:	-26,35 kNm (100 %)
u displacement:	39,01 mm (100 %)	u displacement:	39,01 mm (100 %)
w displacement:	-9,34 mm (24 %)	w displacement:	-9,76 mm (25 %)

Axial and shear stresses

σ x:	-179,46 MPa
τ_max:	0,65 MPa
σ x:	176,56 MPa

Figure 23 Bending moment diagram and detail results

should be noted that this diagram is *always* drawn on the *tension* side of the member. The only result accompanying the diagram is the maximum moment value (26,44 kNm) not shown in figure 23. However, by (left and right) clicking an element, results at the element's ends are shown in a result box as shown in figure 23. Clicking the σ/τ button in this box will, as shown, produce another box with the axial stress (σ_x) at the extreme cross section fibers as well as the maximum shear stress (τ). The latter is not available for cross sections of the arbitrary category.

All individual diagrams can be scaled up or down by using the arrow buttons in the toolbox; they can be normalized again using the N button. For the displacement diagram the toolbox provides a button (Tδ) that will show the displacements with "true" (real) size.

To the left in the ribbon (figure 22) are two more buttons: Reaction forces and Resultants. Clicking the first will produce a view of the model with arrows indicating all non-zero reactions; right-clicking the joint symbol will produce a box with the values of the reaction forces. Right-clicking any joint will produce the residual forces at the joint; for an unsupported joint with no displacement releases ("hinge") these forces should be zero. At a hinge the residual forces are the "hinge forces", the sum of which should be zero for all members at the joint.

The Resultants button will produce the sum of all external loading in the two global directions as well as the sum of all reaction forces in the same directions.

Linearized buckling analysis

Figure 24 shows the ribbon for linearized buckling analysis. The Run analysis arrow

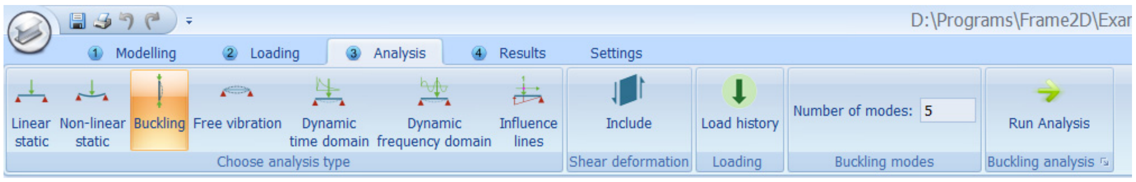
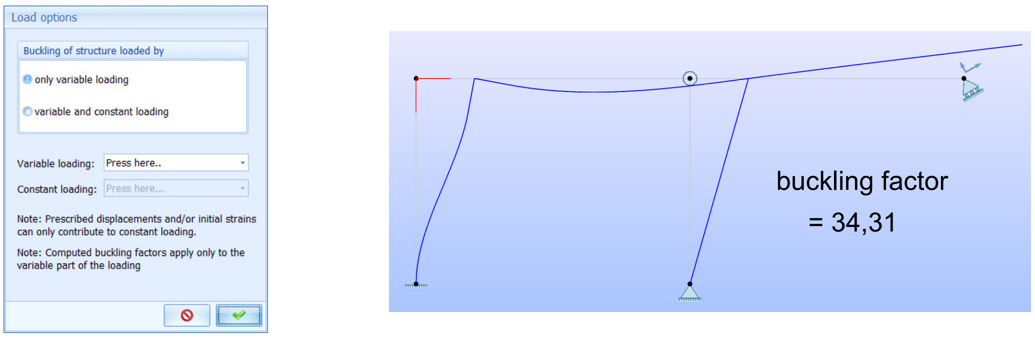


Figure 24 Analysis ribbon for linearized buckling analysis

is active, but unless you have made a visit to the Load history button you most likely will receive an error message telling you that the buckling analysis is aborted due to very low level of compression forces. Load history launches the dialog box shown in figure 25a. Here you will have to make some assumptions and then specify the loading. You can assume all loading to be variable or you can assume both variable and constant loading; the significance of this choice is explained below, but the upshot is that the computed *buckling factors* only apply to the variable loading. Having made


Load options

Buckling of structure loaded by

- [✓ only variable loading]
- variable and constant loading

Variable loading: Press here..

Constant loading: Press here...

Note: Prescribed displacements and/or initial strains can only contribute to constant loading.

Note: Computed buckling factors apply only to the variable part of the loading

(**a**) Load option dialog box (**b**) Typical buckling mode (mode #2)

Figure 25 Load options (a) and typical buckling mode (b)

this choice, the spatial load combination(s) need(s) to be specified. The number of modes may be specified in the ribbon (5 is the default choice) as may inclusion of shear deformations.

Computational aspects

The total loading, $\mathbf{R}$, may consist of a *constant* part, $\mathbf{R}_c$, and a *variable* part, $p\mathbf{R}_v$. The variable part is assumed to vary proportionally with a multiplier p, that is

$$\mathbf{R} = \mathbf{R}_c + p\mathbf{R}_v \tag{1}$$

where $\mathbf{R}_v$ is the *nominal* part of the variable loading (expressed by a spatial load combination). It should be noted that the variable part, $\mathbf{R}_v$, can *only contain external loading* (no prescribed displacements or initial strains).

Linearized buckling analysis is concerned with the 2nd order stiffness matrix

$$\mathbf{K}_2 = \mathbf{K}_m - \mathbf{K}_G(P) \tag{2}$$

where $\mathbf{K}_m$ is the *material* stiffness and $\mathbf{K}_G$, which is a function of the axial forces P, is the *geometric* stiffness. The minus sign in equation (2) assumes the axial forces taken positive as compression (which is a common convention in buckling analysis). The material stiffness is identical to the ordinary 1st order stiffness $\mathbf{K}_0$, modified with respect to bi-linear bar elements. Shear deformations may be included in $\mathbf{K}_0$.

The geometric stiffness matrix may be expressed as

$$\mathbf{K}_G = \mathbf{K}_{Gc} + p\mathbf{K}_{Gv} \tag{3}$$

where $\mathbf{K}_{Gc}$ is the geometric stiffness due to the axial forces (P_c) caused by $\mathbf{R}_c$ acting alone, and $\mathbf{K}_{Gv}$ is the geometric stiffness due to the axial forces (P_v) caused by the nominal $\mathbf{R}_v$ acting alone. Hence

$$\mathbf{K}_2 = \mathbf{K}_0 - \mathbf{K}_{Gc} - p\mathbf{K}_{Gv} = \mathbf{K}_1 - p\mathbf{K}_{Gv} \tag{4}$$

where $\mathbf{K}_1$ is the material stiffness modified with respect to the geometric stiffness effects of the constant part of the loading, that is

$$\mathbf{K}_1 = \mathbf{K}_0 - \mathbf{K}_{Gc} \tag{5}$$

Buckling is now defined as a state for which $\mathbf{K}_2$ becomes singular. For a singular $\mathbf{K}_2$ the homogeneous system of equations

$$\mathbf{K}_2\mathbf{q} = (\mathbf{K}_1 - p\mathbf{K}_{Gv})\mathbf{q} = \mathbf{0} \tag{6}$$

has non-trivial solutions (p_i,$\mathbf{q}_i$). Equation (6) represents a general, symmetric *eigen-problem*. This problem is (by default) solved by so-called *subspace iteration*. However, in the lower right-hand corner of the Run analysis button the small arrow will launch a dialog box from which it is possible to choose a truncated Lanczos method for the eigenvalue extraction.

The separation of the loading into a constant ($\mathbf{R}_c$) and a variable part ($\mathbf{R}_v$), which is controlled by the user (see figure 25a), may be useful in many practical situations where certain loading is always constant (e.g. dead load). The buckling factor then indicates by how much the variable part of the loading can be increased before the structure becomes unstable, which is normally the most interesting question.

Typical results

The only results from a buckling analysis are the buckling mode shapes and the corresponding buckling factor (which is the factor by which the variable part of the loading must be multiplied in order to cause the structure to "buckle" in the corresponding mode shape). In figure 25b is shown the 2nd buckling mode for the frame of figure 17 subjected to the loading of figure 20 (which is all variable). The first buckling mode is simple "Euler buckling" of column B-D, for which the buckling factor is somewhat smaller than that in figure 25b, namely 31,14.

Nonlinear static analysis

Figure 26 shows the ribbon for linearized buckling analysis. The Run analysis is

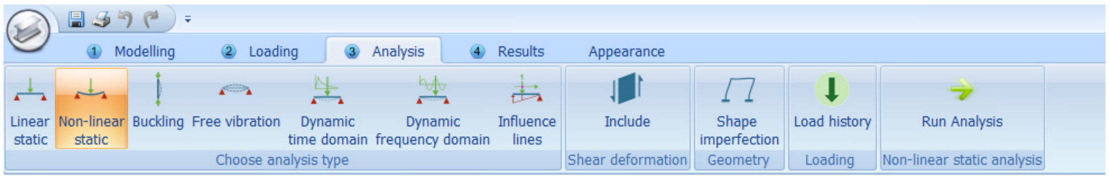
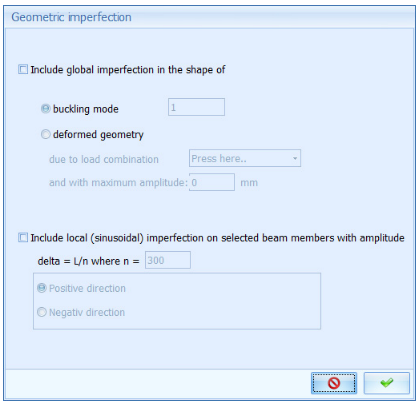


Figure 26 Analysis ribbon for nonlinear static analysis

active, but if the user push it he/she will get a message telling him/her that loading needs to be specified. However, the user may first consider to impose some kind of geometrical imperfection; default is no such imperfections. The shape imperfection button launches the dialog box in figure 27a. A *global* shape imperfection may be



Geometric imperfection

☐ Include global imperfection in the shape of

● buckling mode 1

○ deformed geometry

due to load combination Press here..

and with maximum amplitude: 0 mm

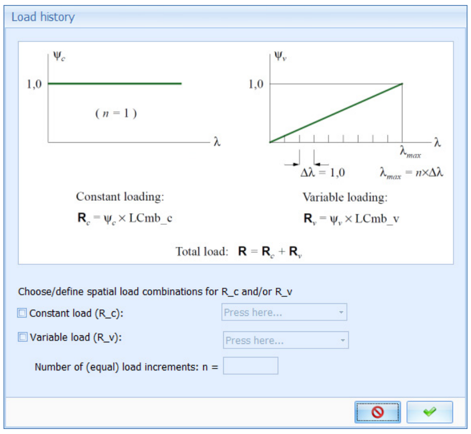
☐ Include local (sinusoidal) imperfection on selected beam members with amplitude

delta = L/n where n = 300

● Positive direction

○ Negativ direction

(**a**)



Load history

Constant loading:

$\mathbf{R}_c = \psi_c \times$ LCmb_c

Variable loading:

$\mathbf{R}_v = \psi_v \times$ LCmb_v

Total load: $\mathbf{R} = \mathbf{R}_c + \mathbf{R}_v$

Choose/define spatial load combinations for R_c and/or R_v

☐ Constant load (R_c): Press here...

☐ Variable load (R_v): Press here...

Number of (equal) load increments: n =

(**b**)

Figure 27 Dialog boxes for geometric imperfection (a) and load history (b)

imposed, the form of which may be in the shape of a buckling mode shape due to a specific load combination or in the shape of the static deformations due to a specified load combination. In either case the user must provide the maximum "amplitude" (in mm). Instead of, or in addition to, the global imperfection, *local* imperfections in the shape of half sine waves (with amplitudes equal to *L*/*n*) may be specified for selected beam (compression) members.

With reference to the dialog box in figure 27b two "types" of loading may be specified, a *constant loading* ($\mathbf{R}_c$) applied in full at "time" zero ($\lambda = 0$) and a *variable* loading ($\mathbf{R}_v$) applied gradually, from zero to full magnitude, in a certain number (= *n*) of *equal* increments. The user may specify the one or the other, or both in combination. In the latter case some of the loading is applied at time zero, and maintained constant throughout, whereas the rest of the loading is applied gradually, in equal increments. Both types of loading are defined in terms of a spatial load combination which may already be defined or it may be defined by selecting Add/edit in the appropriate drop down menu.

Computational aspects

A fairly general method of *nonlinear geometric analysis*, referred to as a *co-rotated formulation* with a consistent tangent stiffness, is used. This theory, which assumes small strains but accounts for large displacements, is more general than so-called 2nd order theory (which is basically based on small displacements). For instance, as opposed to 2nd order theory, the method used by **fap2D** will activate the “hammock” effect (which will produce a large axial force in a simply supported beam subjected to transverse loading only, provide the supports are prevented from horizontal movements).

It should be emphasized that linear, elastic materials are assumed.

The loading, which may consist of external forces, initial strains (*e.g.* temperature) and prescribed displacements, may, as explained above, be applied in one step or in a number of equal increments. In the latter case, individual *response parameters* may be determined (sampled) after each load step, resulting in a response history for each parameter. It should be noted that if a step-wise loading procedure is chosen, part of the loading may be kept constant during the loading process. Hence, the program provides the three loading options shown in figure 28. $\mathbf{R}_c$ and $\mathbf{R}_v$ are the load vectors

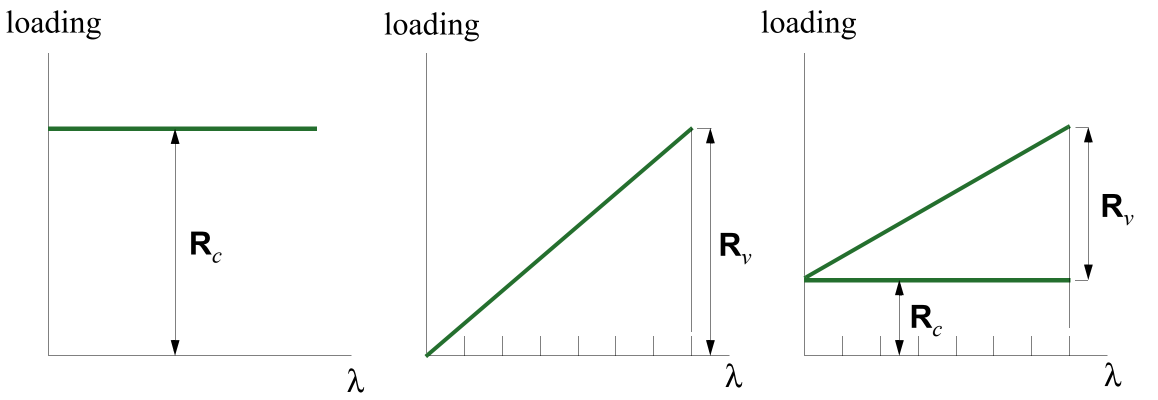


Figure 28 Loading options for nonlinear static analysis

corresponding to spatial load combinations. It should be noted that the variable part of the loading ($\mathbf{R}_v$) must consist of external loads and/or prescribed displacements (but *not* initial strain); the constant part of the loading ($\mathbf{R}_c$) may contain all types of loading.

If more loading than the structure can sustain is applied, the subroutine will (automatically) reduce and adjust the loading to a level within 1% of the “maximum” loading the structure can support before it becomes unstable. This needs an explanation since “maximum loading” is, to some extent, dependent on how the loading is applied.

If all loading is applied in *one* step, then depending on the magnitude of the loading, one of two things can happen:

1. The first version of the *tangent stiffness matrix* is based on the 1st order linear stiffness matrix $\mathbf{K}_0$ and the geometric stiffness matrix $\mathbf{K}_G$ is based on the axial member forces obtained by solving the problem

$$\mathbf{K}_0\mathbf{r}_0 = \mathbf{R}_c \tag{7}$$

If the loading Rc is sufficiently large, the matrix $\mathbf{K}_0 - \mathbf{K}_G$ may be indefinit, and this is taken as an indication that $\mathbf{R}_c$ is more than the structure can support. The loading is halved and the procedure starts all over again. If this results in a positive definite matrix, the program will iterate until the unbalanced forces are sufficiently small, and a new load increment, which is half of the previous loading is applied. A new tangent matrix is established and tested for positive definitness. If positive definite equilibrium iterations are carried out and a new load increment, which is half of the previous one is applied; if not positive definite the program backs up to the last equilibrium position and halve the load increment once again before the procedure is repeated. This goes on until an equilibrium position is obtained for a loading within 1% of the smallest (total) load that cause an indefinit tangent stiffness matrix. It follows that this loading must be *smaller* than the loading $\mathbf{R}_c$ originally applied.

2. If the loading $\mathbf{R}_c$ does not cause the tangent stiffness matrix to become negative definite, the program will carry out equilibrium iterations with full loading on the structure, which involves updating the geometry, until the unbalanced forces are sufficiently small.

If the same total loading is applied as a variable load in, say 20 equal increments, the structure may well be capable of sustaining considerable more load than if the same load was applied in just one step. This will be demonstrated by a simple example in the next section, and it has to do with the fact that gradual loading facilitates force redistribution to take place which in turn may give to structure more apparent strength. Apparent because material failure will most likely occur long before the “maximum stability load” is attained. This discussion is therefore of more academic than practical interest.

Typical results

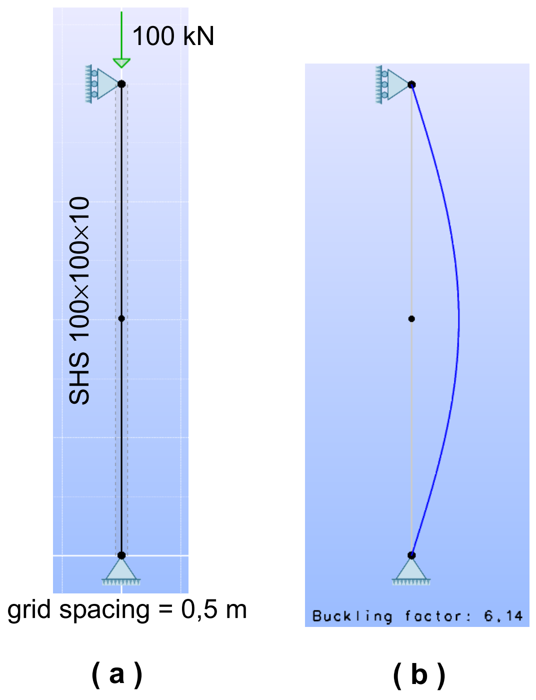


Figure 29 A simple EULER column

The results available from a nonlinear static analysis are the same as those obtained by a linear analysis, except that a nonlinear analysis may also provide “time history” plots for specified response parameters. The latter may give useful information about the nonlinearity of the problem.

The simply supported (EULER-) column in figure 29a serves as an example. A linearized buckling analysis suggests a buckling load of 614 kN (figure 29b).

Next we increase the load to 1000 kN and carry out a nonlinear analysis with a geometrical imperfection with the shape of the first buckling mode (figure 29b) and an amplitude of 16 mm (= $L / 250$).

First we apply all loading at once (only constant loading). The program comes back with the message that 746 kN is within 1% of the load the column can support before it becomes unstable. In other words starting load of 1000 kN clearly caused a negative definite tangent stiffness. The results from the analysis are shown in figure 30.

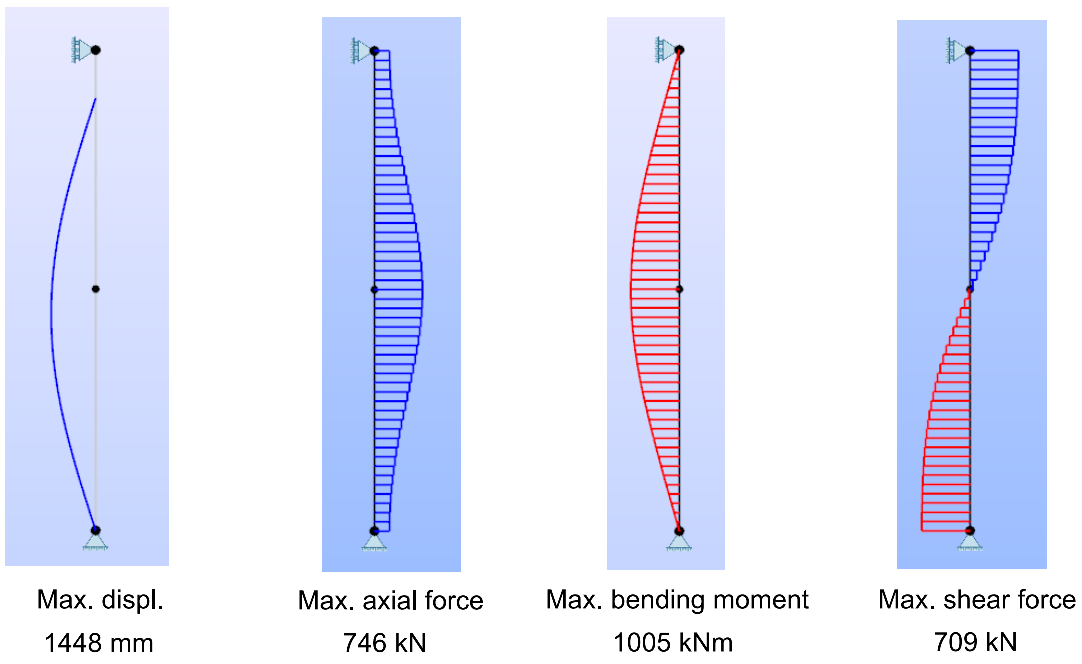


Figure 30 Results for the EULER column subjected to 1000 kN applied at once

We repeat the analysis, but this time we apply the loading gradually, in 50 equal increments. This time results, shown in figure 31, are obtained for the full load (1000 kN).

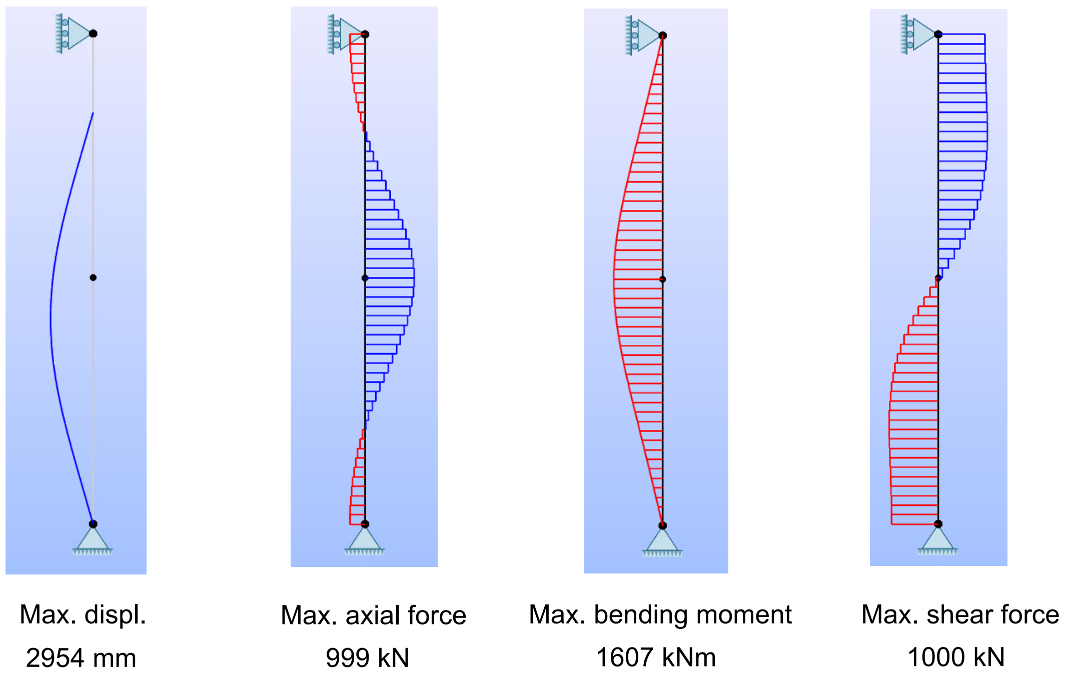
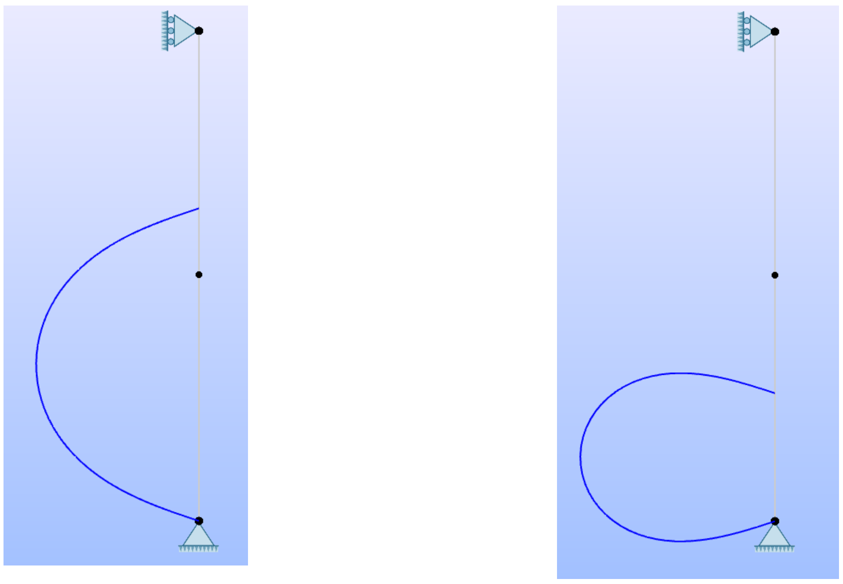


Figure 31 Results for the EULER column subjected to 1000 kN applied incrementally (50 load increments)

When comparing results it should be kept in mind that they are normalized results. This is particularly important for the displacement which clearly is not drawn to scale in figures 30 and 31. If we, instead of Show all click the Displacement button and then the Tδ (true displacement) button in the toolbox, a completely different picture appears, see figure 32. And now the tensile axial forces in figure 31 make sense.



(**a**) P = 746 kN (fig. 30) (**b**) P = 1000 kN (fig. 31)

Figure 32 True displacements

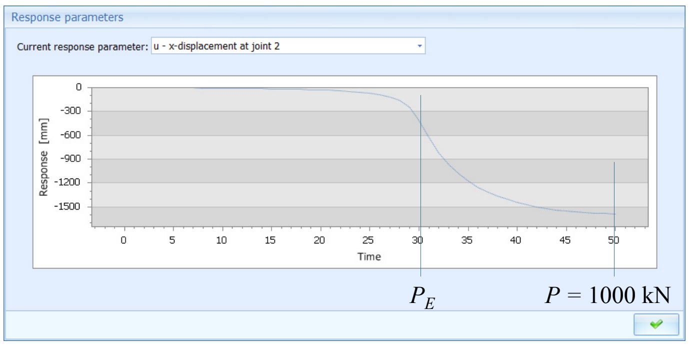
The results of figures 30, 31 and 32 are of more academic than practical interest. On closer inspection we find that stresses are of magnitude 15 000 MPa; hence material failure will have occurred long before the displacements of these figures are attained.

The lesson here is that loads that can possibly vary should preferably be applied incrementally.

Another useful result available after a nonlinear analysis is the “time history” of defined *response parameters*. Figure 33 shows how the horizontal displacement of the mid-point of the column varies with “time” λ (which is really a measure of the external load), for the loading case of figure 31. This type of result requires (a) that

Figure 33

Horizontal displacement
vs
“tilme” or loading



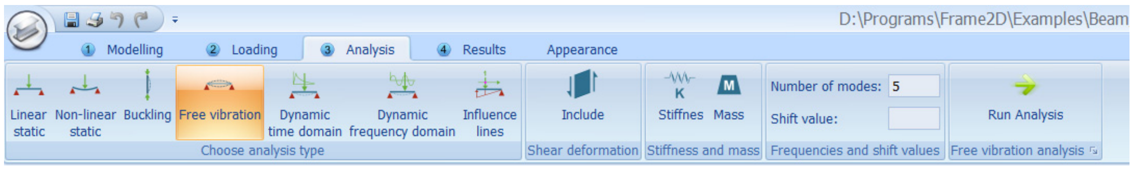
Response parameters

Current response parameter: u - x-displacement at joint 2

response parameters have been defined and (b) that the load is applied incrementally, preferably with a significant number of load increments.

Free, undamped vibration analysis

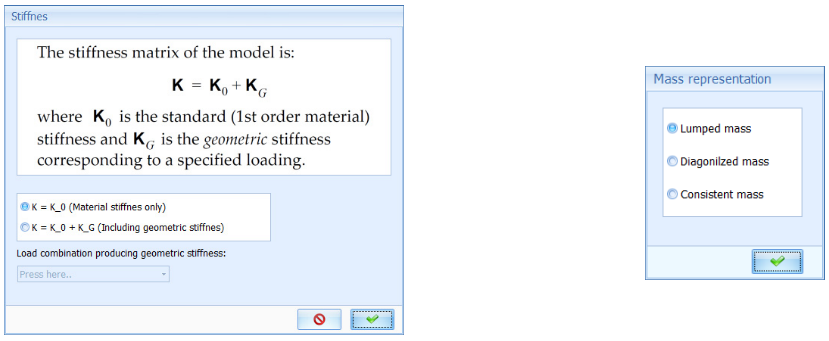
Figure 34 shows the ribbon for free vibration analysis. The Run analysis arrow is



Number of modes: 5
Shift value:

Figure 34 Analysis ribbon for free vibration analysis

active, and if pushed a free vibration analysis will be carried out for default choices for stiffness and mass. Figure 35 shows the dialog boxes "behind" the Stiffness and Mass buttons. We see that it is possible to modify the structural stiffness by including



Stiffnes

The stiffness matrix of the model is:

$$\mathbf{K} = \mathbf{K}_0 + \mathbf{K}_G$$

where $\mathbf{K}_0$ is the standard (1st order material) stiffness and $\mathbf{K}_G$ is the *geometric* stiffness corresponding to a specified loading.

- [✓ K = K_0 (Material stiffnes only)]
- K = K_0 + K_G (Including geometric stiffnes)

Load combination producing geometric stiffness:

Press here..

Mass representation

- [✓ Lumped mass]
- Diagonilzed mass
- Consistent mass

Figure 35 Stiffness and mass settings for free vibration analysis

the geometric stiffness, due to a particular load combination, before carrying out the free vibration analysis, see next section. An example of this would be to determine the free vibration characteristics of a structure subjected to all dead load, e.g. a bridge. For structures with cable members this may make a significant difference. The default choice for the mass representation is the obvious one for the modelling philosophy of **fap2D**.

The user also needs to specify the number of mode shapes to be determined (default is 5) and where on the frequency axis the eigenvalues shall be extracted, defined by the so-called *shift* value σ (default is σ = 0). More about this below.

Computational aspects

The numerical problem to solve is the general, symmetric eigenproblem

$$(\mathbf{K} - \omega^2 \mathbf{M})\mathbf{q} = \mathbf{0} \tag{8}$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{M}$ is the mass matrix, ω is the circular frequency of the free vibration ($\lambda = \omega^2$ is the eigenvalue of the problem) and $\mathbf{q}$ is the corresponding mode of vibration (eigenvector). Normally the program will determine a limited number (n) of the *lowest* eigenvalues and corresponding eigenvectors (mode shapes), so-called eigenpairs ($\lambda_i, \mathbf{q}_i$), that satisfy eqn. (8). If a non-zero shift (σ) is specified, the modified problem

$$(\mathbf{K}_\sigma - \mu\mathbf{M})\mathbf{q} = \mathbf{0} \tag{9}$$

is solved, where $\mu = \lambda - \sigma$ and $\mathbf{K}_\sigma = \mathbf{K} - \sigma\mathbf{M}$. The problem is equivalent to finding n roots of the characteristic polynomial in the vicinity of σ on the frequency axis, see figure 36.

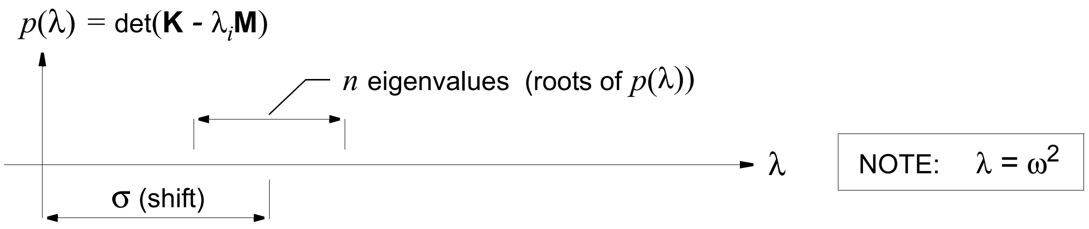

$p(\lambda) = \det(\mathbf{K} - \lambda_i\mathbf{M})$
n eigenvalues (roots of $p(\lambda)$)
λ
NOTE: $\lambda = \omega^2$
σ (shift)

Figure 36 Schematic presentation of the free vibration eigenvalue problem

The eigenvalue problem of eqn. (8) is (by default) solved by so-called *subspace iteration*. However, as for linearized buckling, in the lower right-hand corner of the Run analysis button the small arrow will launch a dialog box from which it is possible to choose a truncated Lanczos method for the eigenvalue extraction.

Results

The only results from a free vibration analysis are the mode shapes and corresponding frequencies (in Hz). Displacement plots very similar to the buckling mode shapes are produced, accompanied by the corresponding frequency. For the frame in figure 17 the first (or lowest) mode shape is very similar to the second buckling mode shown in figure 25b, its frequency is 7,28 Hz.

A useful feature is animation of the free vibration mode shapes; a start and a stop button is available in the toolbox for the result view.

Forced vibration analysis - time domain

Figure 37 shows the ribbon for forced vibration analysis in the time domain. The Run

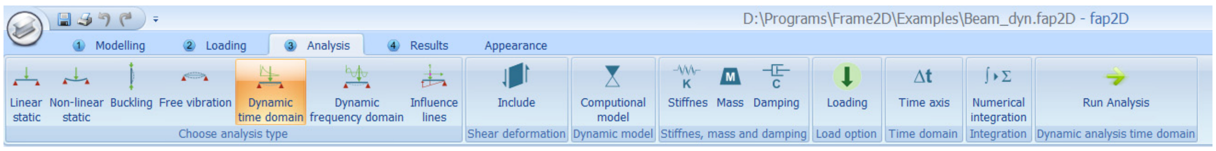
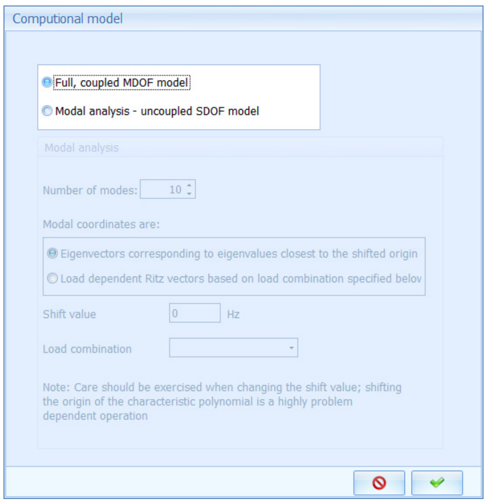


Figure 37 Analysis ribbon for forced vibration analysis in the time domain

analysis arrow is active, but in order to carry out an analysis the user needs to visit at least some of the buttons.

As for all other analyses, the Include shear deformation button is optional; default is no shear deformations. The Computational model button is also optional in that a viable default setting is in place. This button launches the dialog box in figure 38a which shows the default setting. A full and coupled, multi-degree-of-freedom (MDOF) model is default. The alternative is *modal analysis*, leading to a series of decoupled single-degree-of -freedom equations (SDOF). If modal analysis is chosen, we have a choice of modal coordinates: free vibration mode shapes (which are default, see figure 38b) or load dependent Ritz vectors. If the latter is chosen, a spatial


Computional model

[✓ Full, coupled MDOF model]
Modal analysis - uncoupled SDOF model

Modal analysis

Number of modes: 10

Modal coordinates are:

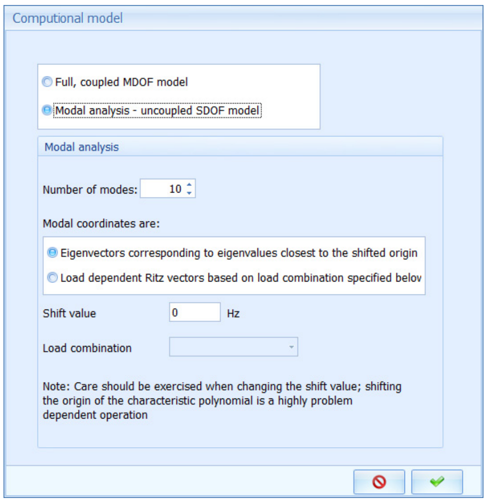
[✓ Eigenvectors corresponding to eigenvalues closest to the shifted origin]
Load dependent Ritz vectors based on load combination specified belov

Shift value 0 Hz

Load combination

Note: Care should be exercised when changing the shift value; shifting the origin of the characteristic polynomial is a highly problem dependent operation

(**a**) default setting


Computional model

Full, coupled MDOF model
[✓ Modal analysis - uncoupled SDOF model]

Modal analysis

Number of modes: 10

Modal coordinates are:

[✓ Eigenvectors corresponding to eigenvalues closest to the shifted origin]
Load dependent Ritz vectors based on load combination specified belov

Shift value 0 Hz

Load combination

Note: Care should be exercised when changing the shift value; shifting the origin of the characteristic polynomial is a highly problem dependent operation

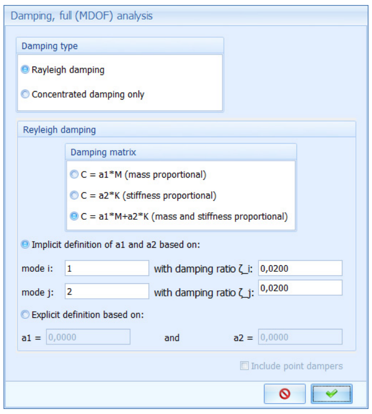
(**b**) modal analysis

Figure 38 Computational model for forced vibration analysis in the time domain

load combination must be specified (as basis for the Ritz vectors).

The Stiffness and Mass buttons serve the same purpose as for free vibration analysis; it is, in other words, possible to include geometric effects in the stiffness matrix also for this type of analysis. The default settings of both buttons are reasonable in most cases.

The Damping button also has a default setting, shown in figure 39a if an MDOF model is specified, and in figure 39b if modal analysis is chosen. Both may serve as reasonable first assumptions. Visiting this button is therefore not a must.


Damping, full (MDOF) analysis

Damping type
[✓ Rayleigh damping]
Concentrated damping only

Reyleigh damping

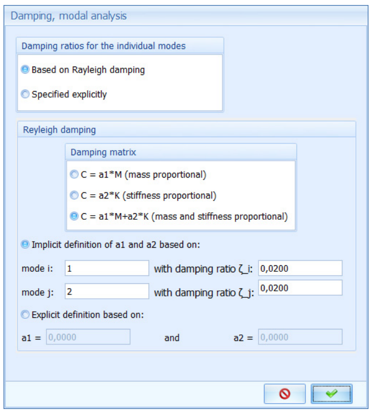
Damping matrix
C = a1*M (mass proportional)
C = a2*K (stiffness proportional)
[✓ C = a1*M+a2*K (mass and stiffness proportional)]

[✓ Implicit definition of a1 and a2 based on:]
mode i: 1 with damping ratio ζ_i: 0,0200
mode j: 2 with damping ratio ζ_j: 0,0200

Explicit definition based on:
a1 = 0,0000 and a2 = 0,0000

Include point dampers

(**a**)


Damping, modal analysis

Damping ratios for the individual modes
[✓ Based on Rayleigh damping]
Specified explicitly

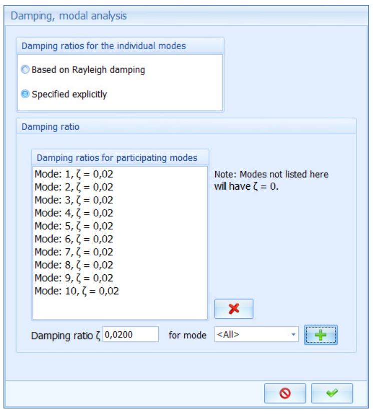
Reyleigh damping

Damping matrix
C = a1*M (mass proportional)
C = a2*K (stiffness proportional)
[✓ C = a1*M+a2*K (mass and stiffness proportional)]

[✓ Implicit definition of a1 and a2 based on:]
mode i: 1 with damping ratio ζ_i: 0,0200
mode j: 2 with damping ratio ζ_j: 0,0200

Explicit definition based on:
a1 = 0,0000 and a2 = 0,0000

(**b**)


Damping, modal analysis

Damping ratios for the individual modes
Based on Rayleigh damping
[✓ Specified explicitly]

Damping ratio

Damping ratios for participating modes
Mode: 1, ζ = 0,02
Mode: 2, ζ = 0,02
Mode: 3, ζ = 0,02
Mode: 4, ζ = 0,02
Mode: 5, ζ = 0,02
Mode: 6, ζ = 0,02
Mode: 7, ζ = 0,02
Mode: 8, ζ = 0,02
Mode: 9, ζ = 0,02
Mode: 10, ζ = 0,02

Note: Modes not listed here will have ζ = 0.

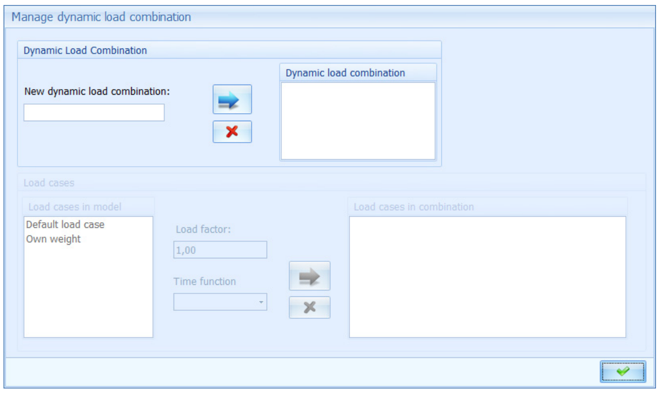
Damping ratio ζ 0,0200 for mode <All>

(**c**)

Figure 39 Damping models

It should be emphasized that all damping models assume *viscous* damping. For modal analysis we see that the damping ratios (ζ_i) may be computed on the basis of Rayleigh damping and assumed damping ratios of two eigenmodes (to be specified if not the first and second), or they may be given explicitly for each modal coordinate, as in figure 39c (where the All choice has been used to assign the default value of 0,02).

Next we come to the Loading button, and this button *must* be used on the first visit to this ribbon. The associated dialog box is shown in figure 40. Here we need to define



New dynamic load combination:

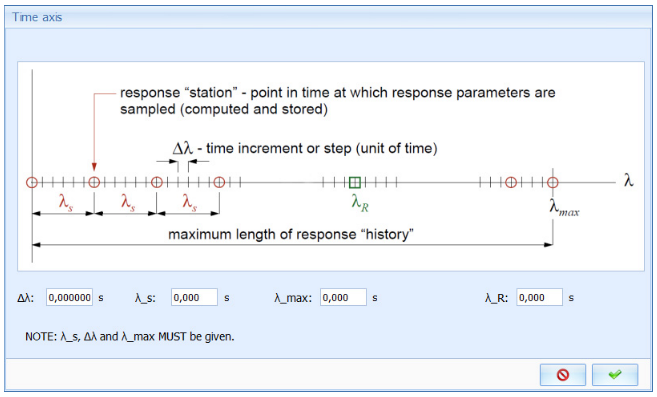
Default load case
Own weight

Load factor: 1,00

Figure 40 Dynamic load combination, for forced vibration in the time domain

at least one dynamic load combination, and this combination is made up of one or more combinations of a spatial load case and a time function. If the appropriate load case and/or time function is/are not available at this stage, the user needs to go back to the loading ribbon and define the missing items.

A visit to the Time axis button is also a must. The corresponding dialog box is shown in figure 41. We need to give the three first parameters, $\Delta\lambda$, λ_s and λ_{max}, all in seconds.


Time axis
response "station" - point in time at which response parameters are sampled (computed and stored)
Δλ - time increment or step (unit of time)
λ
λs
λs
λs
λR
λmax
maximum length of response "history"

Δλ: 0,000000 s λ_s: 0,000 s λ_max: 0,000 s λ_R: 0,000 s

NOTE: λ_s, Δλ and λ_max MUST be given.

Figure 41 The time axis for forced vibration in the time domain

The fourth parameter, λ_R, defines a point in the relevant time domain, at which a complete response (nodal displacements, element section forces and reaction forces) is computed and made available for diagrammatic presentation. $\Delta\lambda$ is a key parameter here, and we shall return to this quantity and make some comments about it in the next two sections.

The last button in the ribbon of figure 37 is concerned with Numerical integration. Viable default values are provided, and we will therefore leave it for the moment; we return to it in the next section.

Computational aspects

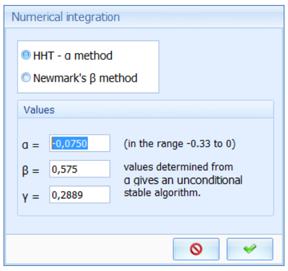
The equation of motion for the MDOF model

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{R}(t) \tag{10}$$

or the decoupled equation

$$\ddot{x}_i + 2\varsigma_i\omega_i\dot{x}_i + \omega_i^2 x_i = \bar{R}_i(t) \tag{11}$$

for the modal (SDOF) system, are both solved by *implicit numerical integration*. Without going into details, NEWMARK's β-method and the modified HHT-α method (due to HILBER, HUGHES and TAYLOR) are available for the integration task. NEWMARK's method is governed by two parameters, β and γ, which together with the size of the time step, Δ*t*, define the variation of the acceleration over a time step, the stability of the solution, the amount of algorithmic damping and the accuracy of the method. The most commonly used values are: γ = 0,5 and β = 0,25 (constant average acceleration over the time step), for which the method is unconditionally stable. For γ = 0,5 and β = 0,16667 the method describes linear acceleration (which is only conditionally stable).


Numerical integration

- [✓ HHT - α method]
- Newmark's β method

Values

α =	-0,0750	(in the range -0.33 to 0)
β =	0,575	values determined from α gives an unconditional stable algorithm.
γ =	0,2889	

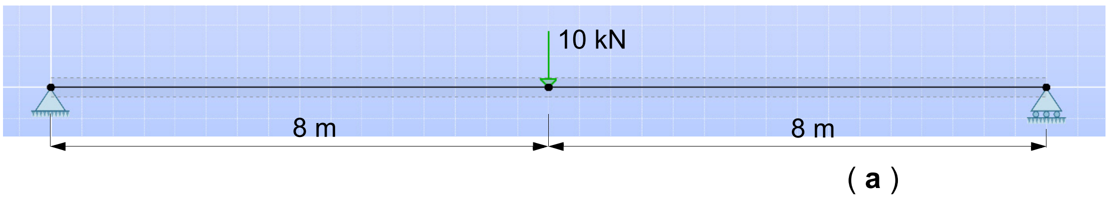
Figure 42
Numerical integration

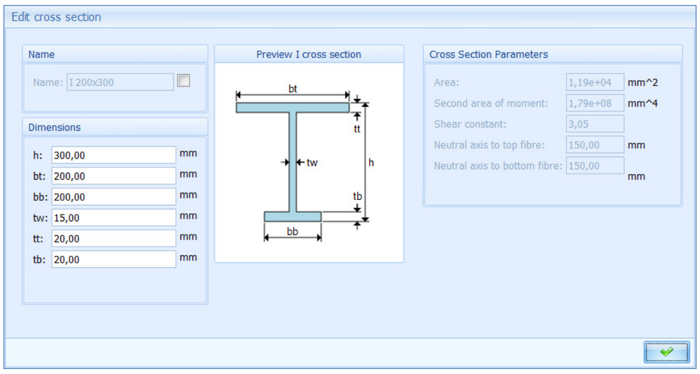
In the HHT-α method, a third parameter α is introduced in addition to β and γ. The purpose of this method is to include algorithmic damping of high frequency "noise" without much loss of accuracy.

For $-1/3 \le \alpha \le 0$ and $\gamma = (1\text{-}2\alpha)/2$ and $\beta = (1\text{-}\alpha)^2/4$ the method is unconditionally stable.

Default values for the HHT-α method, which is the default choice, are shown in figure 42.

Typical results




Edit cross section

Name

Name: I 200x300

Dimensions

h:	300,00	mm
bt:	200,00	mm
bb:	200,00	mm
tw:	15,00	mm
tt:	20,00	mm
tb:	20,00	mm

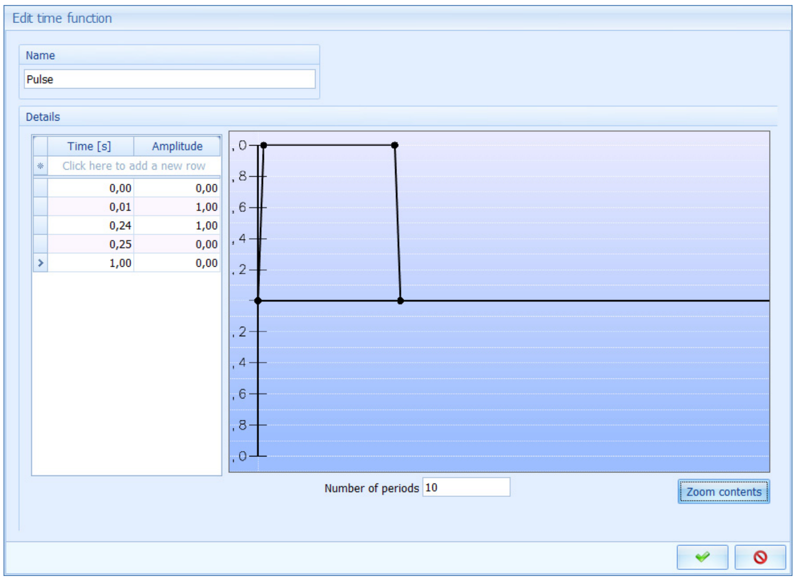
Cross Section Parameters

Area:	1,19e+04	mm^2
Second area of moment:	1,79e+08	mm^4
Shear constant:	3,05	
Neutral axis to top fibre:	150,00	mm
Neutral axis to bottom fibre:	150,00	mm

(**b**) beam cross section

Figure 43 Example problem

Before pushing the Run analysis button, make sure that at least one *response parameter* has been defined. In order to demonstrate the capabilities of this type of analysis we consider a very simple problem, namely the simply supported steel beam shown in figure 43a. The beam has a parametric cross section shown in figure 43 b, and it is subjected to a vertical point load of 10 kN at the middle of the beam. The time vari-


Edit time function

Name: Pulse

Details

Time [s]	Amplitude
0,00	0,00
0,01	1,00
0,24	1,00
0,25	0,00
1,00	0,00

Number of periods: 10

Figure 44 Time function “Pulse”

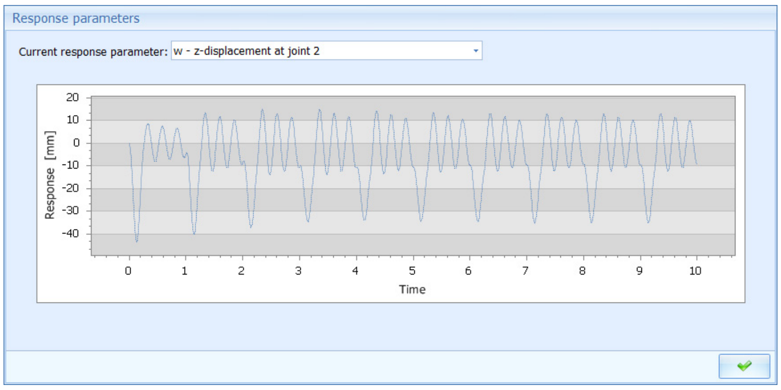
ation of the load is a “pulse” of 1,0 sec duration shown in figure 44. Ten consecutive periods of this pulse is considered as the (total) time function.

Back to the time axis parameters, see figure 41, we now (somewhat arbitrarily) specify the time increment to be $\Delta\lambda = 0{,}001$ s. We sample the response parameter, the vertical displacement of the loaded mid-point, at each 5th time increment, that is $\lambda_s = 0{,}005$ s. The duration of the response range is set equal to that of the loaded period, hence $\lambda_{max} = 10{,}0$ s. The last parameter, λ_R, we leave alone, and its value of zero indicates no system results. With these parameters and the default choice for the other buttons, *i.e.*, MDOF computational model, only material stiffness, lumped mass model, Rayleigh damping based on a damping ratio $\zeta = 0{,}02$ in the two lowest natural modes, and an HHT-α integration scheme with the default values of figure 42, the vertical displacement of the beam mid-point as a function of time is shown in figure 45a.

We repeat the analysis with one change: instead of the MDOF model we now use *modal analysis* with the 10 lowest modes of free vibration as modal coordinates. Otherwise the same assumptions as before. The time-history plot for this model is shown in figure 45b.

We repeat the analysis yet again; still modal analysis, but this time we use 10 load dependent Ritz vectors as modal coordinates. The point load is used as loading for the Ritz vectors. The time history plot for this model is shown in figure 45c.

We see that with the resolution of the plots in figure 45 it is impossible to distinguish the three plots. For this simple example that is hardly surprising. Another comment is that the time increment used in this case (0,001s) probably is unnecessarily small. This can easily be established by a simple sensitivity analysis in which results

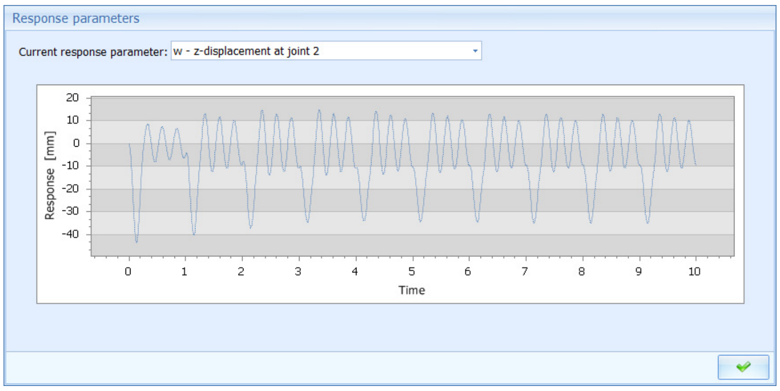


Response parameters

Current response parameter: w - z-displacement at joint 2

(a)

Full, MDOF model

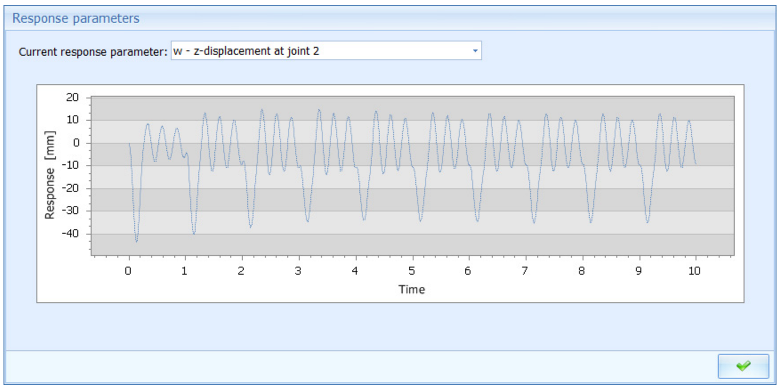


Response parameters

Current response parameter: w - z-displacement at joint 2

(b)

Modal analysis with the 10 lowest free vibration modes as modal cooordinates



Response parameters

Current response parameter: w - z-displacement at joint 2

(c)

Modal analysis with 10 load dependent Ritz vectors as modal cooordinates

Figure 45 Time-history plots of the vertical displacement at the mid-point of the beam

obtained with different values of the time increment (Δλ) are compared. However, with the current computational power of a standard PC, some “overkill” is probably justified.

Forced vibration analysis - frequency domain

Figure 37 shows the ribbon for forced vibration analysis in the time domain. The Run

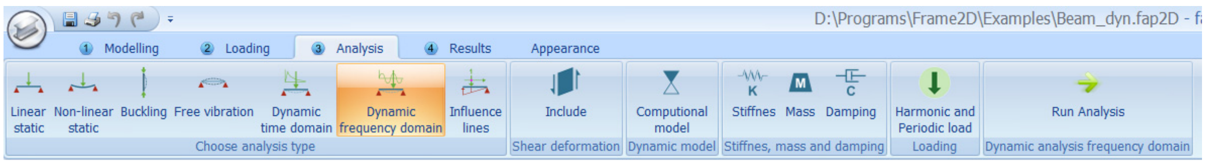


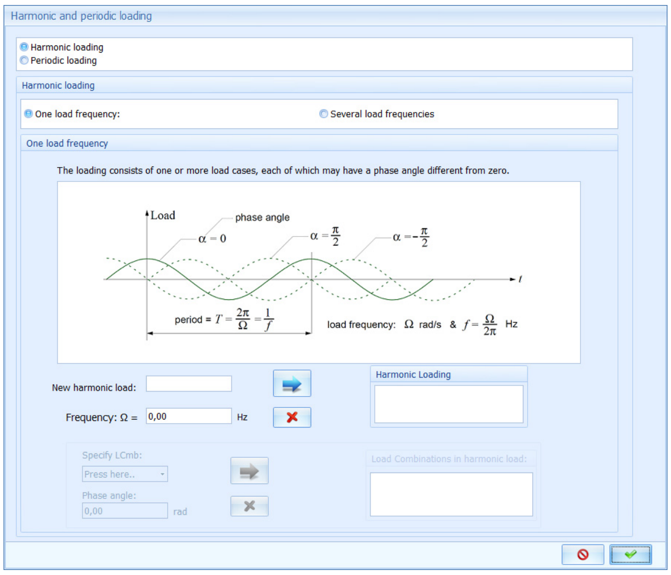
Figure 46 Analysis ribbon for forced vibration analysis in the time domain

analysis arrow is active, but in order to carry out an analysis the user needs to visit the Harmonic and Periodic load button which launches the dialog box in figure 47a. There are three load options in the frequency domain:

1. *One harmonic load combination* where *all* contributing spatial load cases have a harmonic variation with the *same* (load) frequency. However, the various contributing load cases may have *different phase angles*. This is the program's default choice, see figure 47a.
   The results from this analysis are the following *steady state* results for all nodal displacements and element section forces: *amplitude* values of, *dynamic amplification* or dynamic load factors (DLF) and *phase angles*.

2. One spatial load combination with harmonic variation applied for a *series* of frequencies, see figure 47b. The user needs to specify a spatial load combination, a frequency range and the number of frequencies within the range for which *steady state* response of all specified response parameters are computed.
   The results are, for each specified response parameter, frequency plots of: the amplitude values, the dynamic load factors (DLF) and the phase angles.

3. One *periodic*, but non-harmonic load combination, see figure 47c. Here the user need to specify a spatial load combination, a time function defining one period of loading plus some information used by the program when approximating the load by a FOURIER series of harmonic components. Two parameters control the FOURIER series expansion, the maximum number of Fourier terms (default value is 75) and a tolerance parameter (default value is 0,05) that control the actual number of terms used. The user should not change these numbers before an inspection of the time function approximation has been made; this approximation is available after the analysis has been carried out, more below.
   The user also has to provide the number of time increments in the period - the program will determine the (steady state) response at each time increment.
   The main results provided are the steady state response curves over one time period for all specified response parameters; hence, this type of analysis is not relevant unless at least one response parameter has been specified (at a joint). The only other piece of information available after a successful analysis is a plot of the (FOUIER) approximation of the time function.



Harmonic and periodic loading

- [✓ Harmonic loading]
- Periodic loading

Harmonic loading

[✓ One load frequency:] Several load frequencies

One load frequency

The loading consists of one or more load cases, each of which may have a phase angle different from zero.

Load
phase angle
$\alpha = 0$
$\alpha = \frac{\pi}{2}$
$\alpha = -\frac{\pi}{2}$
t
period = $T = \frac{2\pi}{\Omega} = \frac{1}{f}$
load frequency: Ω rad/s & $f = \frac{\Omega}{2\pi}$ Hz

New harmonic load:

Frequency: Ω = 0,00 Hz

Harmonic Loading

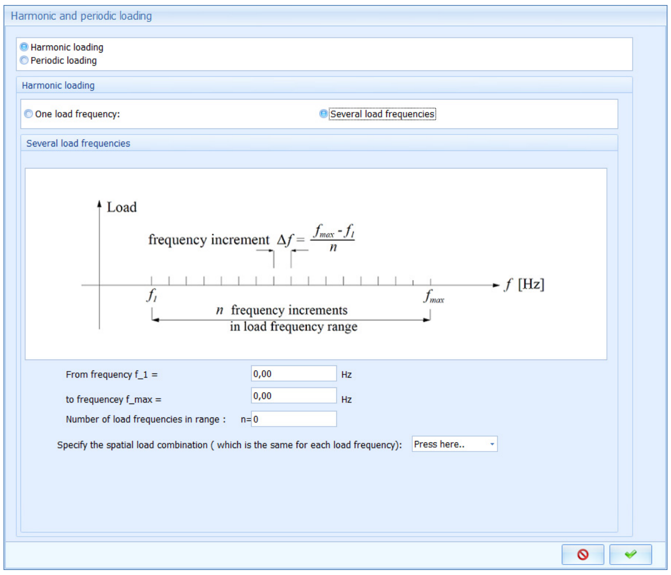
Specify LCmb: Press here..

Phase angle: 0,00 rad

Load Combinations in harmonic load:

(a)

Harmonic loading with ***one*** load frequency



Harmonic and periodic loading

- [✓ Harmonic loading]
- Periodic loading

Harmonic loading

One load frequency: [✓ Several load frequencies]

Several load frequencies

Load
frequency increment $\Delta f = \frac{f_{max} - f_1}{n}$
f [Hz]
f_1
f_{max}
n frequency increments in load frequency range

From frequency f_1 = 0,00 Hz

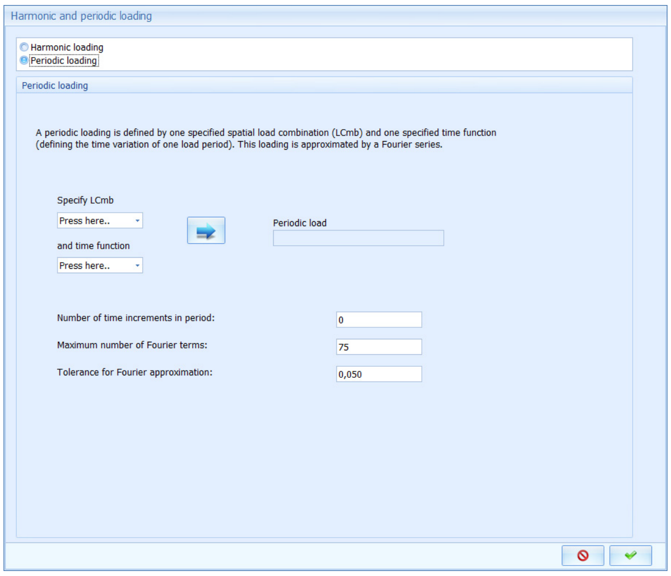
to frequencey f_max = 0,00 Hz

Number of load frequencies in range : n= 0

Specify the spatial load combination (which is the same for each load frequency): Press here..

(b)

Harmonic loading with ***several*** load frequencies



Harmonic and periodic loading

- Harmonic loading
- [✓ Periodic loading]

Periodic loading

A periodic loading is defined by one specified spatial load combination (LCmb) and one specified time function (defining the time variation of one load period). This loading is approximated by a Fourier series.

Specify LCmb: Press here..

and time function: Press here..

Periodic load

Number of time increments in period: 0

Maximum number of Fourier terms: 75

Tolerance for Fourier approximation: 0,050

(c)

Periodic

non-harmonic loading

Figure 47 Load options in the frequency domain

Computational aspects

The problem is solved by the *frequency response method* using, as for the time domain,

- the full, coupled multi-degree-of-freedom (MDOF) system,

or

- superposition of a system of decoupled, single-degree-of-freedom (SDOF) equations obtained by modal analysis; both eigenvectors and so-called load dependent Ritz vectors may be used as modal coordinates.

Representation of stiffness, mass and damping is also the same as in the time domain.

The problem is to find the solution of

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \mathbf{K}\mathbf{r} = \mathbf{R}(\Omega) = \tilde{\mathbf{R}}e^{i\Omega t} \tag{12}$$

Ω is the frequency of the applied *harmonic* loading. We seek the particular solution of eqn. (12), that is the so-called *steady state* solution

$$\mathbf{r} = \tilde{\mathbf{r}}e^{i\Omega t} \tag{13}$$

which has the same frequency as the loading. A tilde (~) on top of a symbol denotes a *complex* quantity. Substituting eqn. (13) into eqn. (12) yields:

$$-\Omega^2\mathbf{M}\tilde{\mathbf{r}} + i\Omega\mathbf{C}\tilde{\mathbf{r}} + \mathbf{K}\tilde{\mathbf{r}} = \tilde{\mathbf{R}} \tag{14}$$

or

$$\tilde{\mathbf{K}}\tilde{\mathbf{r}} = \tilde{\mathbf{R}} \tag{15}$$

where

$$\tilde{\mathbf{K}} = \mathbf{K} - \Omega^2\mathbf{M} + i\Omega\mathbf{C} \tag{16}$$

is the *complex dynamic stiffness* matrix. The solution of eqn. (15) gives the complex response vector $\tilde{\mathbf{r}}$. An arbitrary response component (*dof* number *j*) may be expressed as

$$r_j = r_{j0}e^{i(\Omega t + \beta_j)} = \tilde{r}_j e^{i\Omega t} \tag{17}$$

where

$$\tilde{r}_j = r_{j0}e^{i\beta_j} = r_{jR} + ir_{jI} \tag{18}$$

whose real and imaginary components are

$$r_{jR} = r_{j0}\cos\beta_j \qquad \text{and} \qquad r_{jI} = r_{j0}\sin\beta_j \tag{19}$$

Here

$$r_{j0} = \sqrt{r_{jR}^2 + r_{jI}^2} \tag{20}$$

is the *amplitude* value of the response, and

$$\beta_j = \operatorname{atan}\left(\frac{r_{jI}}{r_{jR}}\right) \tag{21}$$

its *phase angle*. In order to determine the *dynamic amplification*, also referred to as the dynamic load factor (DLF), defined as the ratio between dynamic and static response, the static response needs to be solved. This is accomplished by solving eqn. (15) for $\Omega = 0$, that is

$$\mathbf{K}(\mathbf{r}_R^s + i\mathbf{r}_I^s) = \mathbf{R}_R + i\mathbf{R}_I \tag{22}$$

where superscript s designates *static*. Hence

$$\mathbf{K}\mathbf{r}_R^s = \mathbf{R}_R \qquad \Rightarrow \qquad r_R^s \tag{23}$$

and

$$\mathbf{K}\mathbf{r}_I^s = \mathbf{R}_I \qquad \Rightarrow \qquad r_I^s \tag{24}$$

For the j'th component the static response (for the most un favorable position of the "static harmonic wave") is

$$r_{j0}^s = \sqrt{(r_{jR}^s)^2 + (r_{jI}^s)^2} \tag{25}$$

and the dynamic amplification for component j is

$$DLF_j = \frac{r_{j0}}{r_{j0}^s} \tag{26}$$

Typical results

We return to the example problem in figure 43, and we apply the point load as *one harmonic* load with a frequency of $\Omega = 4$ Hz. This is a very simple example of the loading situation of figure 47a. Using the program's default settings for computational model, stiffness, mass and damping, the steady state results obtained for the bending moments in the beam are shown in Figure 48: amplitude values (figure 48a), DLF-values (figure 48b) and phase angles (figure 48c).

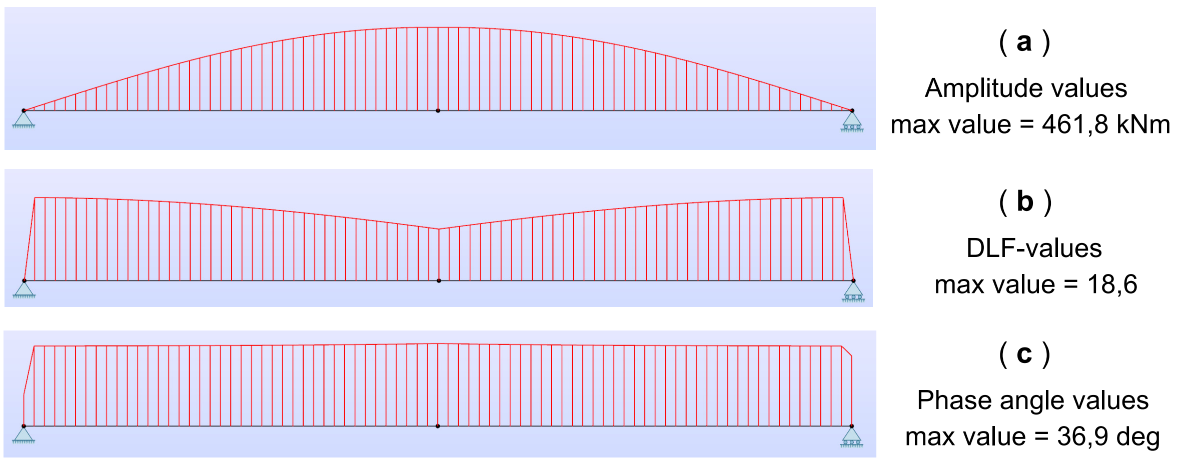
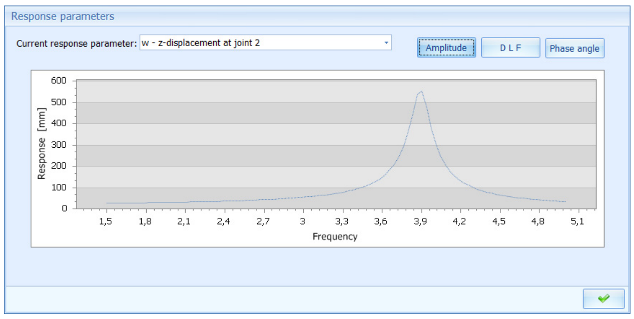


Figure 48 Bending moment results for the beam in figure 43, due to harmonic P (Ω = 4hz)

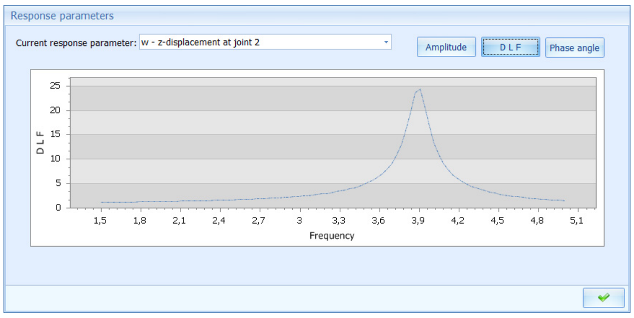
In this case the amplitude diagram in figure 48a may, due to very similar phase angles along the beam, be taken as an actual (and most "severe") bending moment diagram. In general, however, the moment (amplitude) values over the structure will not occur at the same time. We see that the dynamic effects are quite strong - the largest DLF-value is 18,6, for a damping ratio of 0,02 in the first two modes. Keep in mind though that the load frequency is close to the "resonance" frequency of 3,89 Hz. For a load frequency of 3 Hz, the largest DLF-value for the bending moment is 2,85.

The phase angle shows some discrepancies at the ends of the beam; these are believed to be caused by inappropriate handling of the ratio between very small numbers. Results similar to those of figure 48 are available also for displacement, axial force and shear force.

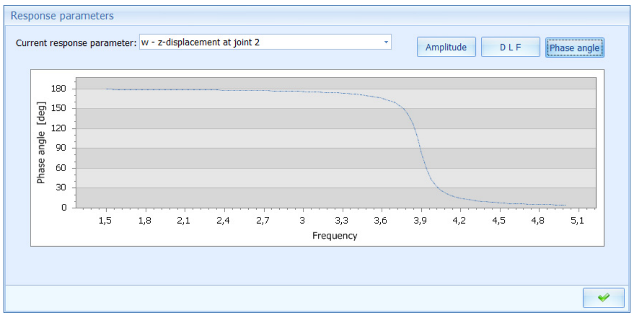
Next, we apply the same point load, but this time as a series of harmonic load cases with 100 different frequencies equally spaced between 1,5 and 5 Hz. This is a simple example of the loading situation of figure 47b. Results will only be available for specified response parameters of which we have one, the transverse displacement under the load. Again we use the program's default settings for computational model, stiffness, mass and damping, and the steady state results obtained for the transverse displacement under the load are shown in Figure 49 as plots along the frequency axis of the amplitude values (figure 49a), DLF-values (figure 49b) and phase angles (figure 49c). As expected, both amplitude and DLF peak at the lowest natural frequency (3,89


Current response parameter: w - z-displacement at joint 2

(**a**) Amplitude


Current response parameter: w - z-displacement at joint 2

(**b**) DLF (dynamic amplification)


Current response parameter: w - z-displacement at joint 2

(**c**) Phase angle

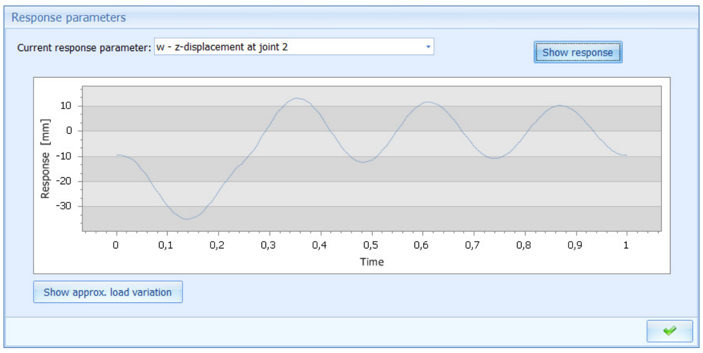
Figure 49 Transverse mid-point displacement due to a harmonic point load P (= 10 kN)

Hz).

Finally we subject the beam to a point load with a *periodic*, but non-harmonic variation in time. The load acts, as before, at the mid-point of the beam, and the time variation of one period (which is 1,0 s long) is the "pulse" shown in figure 44.

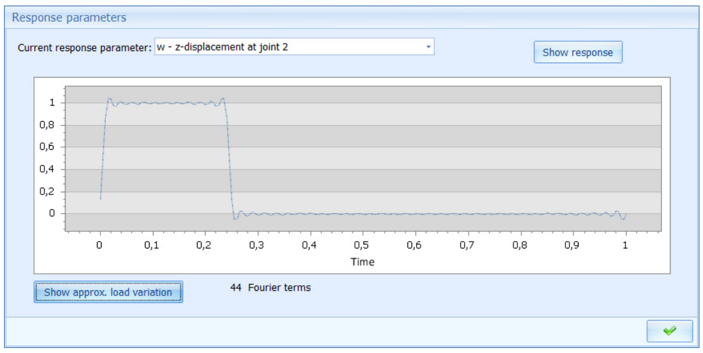
This is a simple example of the loading situation of figure 47c. Again results will only be available for specified response parameters of which we still have defined only one, the transverse displacement under the load. We continue to use the program's default settings for computational model, stiffness, mass and damping. Also for the parameters controlling the FOURIER series approximation the we use the default values (see Figure 47c).

We determine the response at each of 200 equally spaced points in a time period of 1,0 sec. The steady state response of the mid-point displacement over a time period is shown in figure 50a. We see that this plot compares well with the last period of the



(a)

One period of
steady state response



(b)

FOURIER series
approximation of
time function

Figure 50 Periodic, non-harmonic analysis;
response (a) and time function approximation (b)

time domain analysis shown in figure 45 where 10 periods of the same loading were analyzed as a time series. By clicking the button in the left-hand corner of the result box in figure 50a we get the visualization of the FOURIER series approximation of the time function shown in figure 50b. The tolerance parameter (0,05, see figure 47c) is satisfied by 44 FOURIER terms. We recognize the GIBBS phenomenon at all points of abrupt change in the time function.

Influence lines and extreme response

Figure 51 shows the ribbon for analysis of influence lines. The Run analysis arrow is

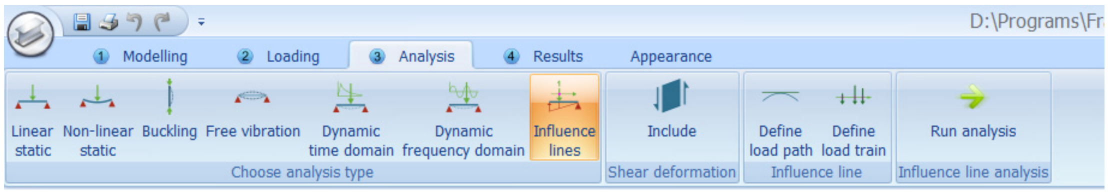
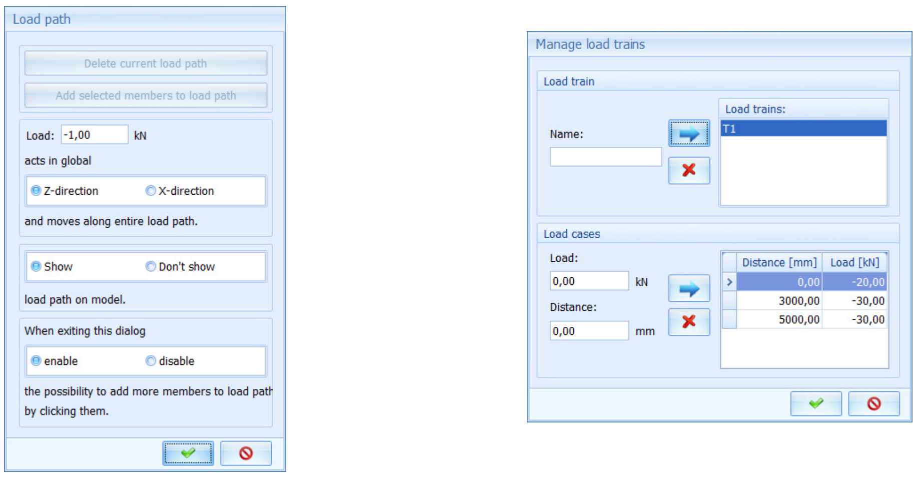


Figure 51 Analysis ribbon for analysis of influence lines

active, but in order to carry out an analysis the user needs to define a *load path* and one or more response parameters must have been or be defined. Since influence lines are normally used to determine the maximum response to loading “moving” accross the load path, such loading, here called a *load train*, should also be defined prior to the actual analysis.

Figure 52 shows the dialog boxes launched by the Define load path and the Define load train buttons, respectively. The load path consists of the members on which the



Load path

Delete current load path

Add selected members to load path

Load: -1,00 kN

acts in global

[✓ Z-direction] X-direction

and moves along entire load path.

[✓ Show] Don't show

load path on model.

When exiting this dialog

[✓ enable] disable

the possibility to add more members to load path by clicking them.

Manage load trains

Load train

Name:

Load trains:

T1

Load cases

Load: 0,00 kN

Distance: 0,00 mm

Distance [mm]	Load [kN]
0,00	-20,00
3000,00	-30,00
5000,00	-30,00

(**a**) (**b**)

Figure 52 Definition of load path (**a**) and load trains (**b**)

load train can travel. Figure 52a suggests that the load path can be defined by marking the relevant members *before* clicking the Define load path button and then click the Add selected members to the load path button (that will be active if this approach is used), or the Define load path button can be clicked without any members marked (as was the case for the dialog box in figure 52a) and then exit the box with enable and click the relevant members. The dialog box in figure 52a also lets the user define the size and direction of the moving point load that “produce” the influence line(s). In order for the extreme response calculations (described below) to work properly, the *unit* value of the point load should *not* be changed. However, its direction must coincide with that of the loads of the load train(s).

A named *load train* can consist of any number of point loads at arbitrary, but user defined mutual distances, and any number of load trains can be defined. Figure 52b shows a load train, named T1, that consists of three point loads, the first of which has a magnitude of 20 kN whilst the other two both are 30 kN. The distance between the first and the second is 2m whereas the last load follows 3m behind the second one.

Computational aspects

Influence line analysis requires a *completely linear model*. Hence bi-linear members (cables and/or struts) cannot be present in the structural model (if present they must be removed or converted into bar members before an analysis is attempted). The computations consist of a (large) series of linear static analyses, one for each load situation. A load situation in turn consists of the (unit) point load placed at a nodal point of a member of the load path. The number of load situations, i.e. the number of load vectors, is therefore defined by the total number of nodal points in the load path members.

Since the model is linear, the stiffness matrix is formed and factorized *once*, and for each load situation the solution is obtained by forward and backward substitutions (which are "cheap" operations compared with factorization). For each load situation the response parameters are computed and stored for presentation. For a particular response parameter each such computed value is drawn as a scaled line segment perpendicular to the horizontal (vertical) projection ot the travel path, starting from the x- (z-) coordinate of the node in question. The influence line of this particular response parameter is the line drawn through the end points of these scaled lines.

Typical results

As a simple example we consider the continuous beam in figure 53, and we seek the influence lines for the vertical displacement at joint 2, the reaction force at joint 4 and the bending moment over the support at joint 5. The load path is the entire beam

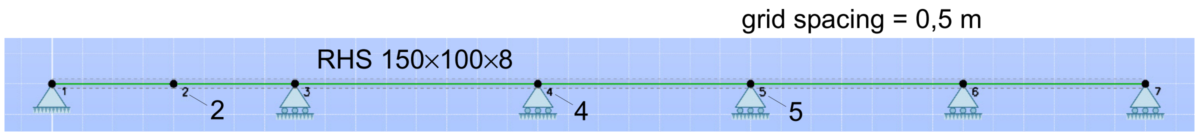
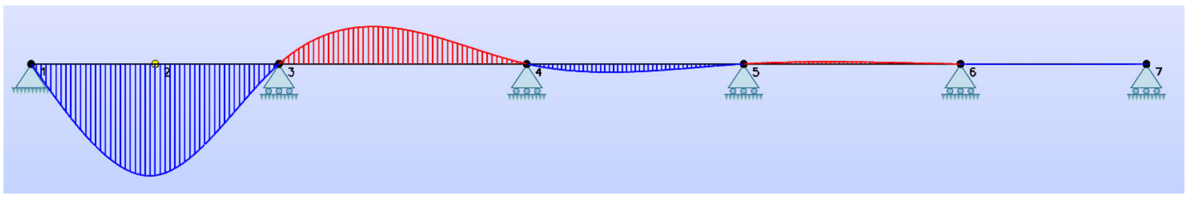


Figure 53 Example problem for influence line analysis

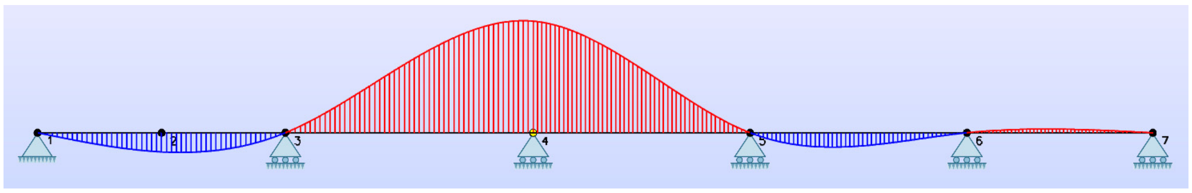
(which is 18 m long), indicated by the green color in the figure.

The influence lines are shown in figure 54a, b and c. The maximum values indicated are those caused by the most "critical" position of the (unit) point load. We see that all influence lines exhibit a value of zero at the supports, except the influence line for the support reaction which is equal (and opposite) to the point load when it is positioned at the support (which make sense).

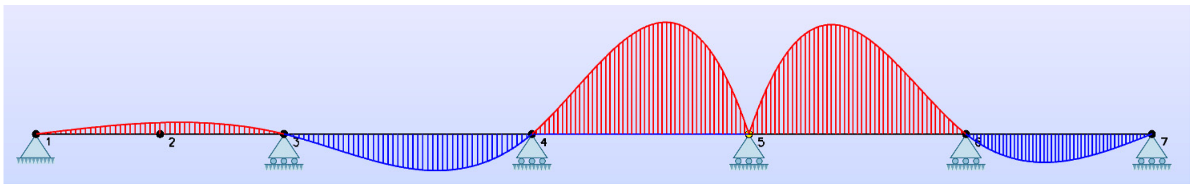
The Result ribbon for influence line analysis is shown in figure 54d. If one or more load trains are defined the Run Xtrm Analysis arrow is active, and pushing the arrow will place the load train at the most "severe" position for the choices made in the ribbon, and a result box will give the value of the response parameter caused by the load train when in this position. Figures 54 e and f show the results for the bending moment at joint 5 when the "train" moves from left to right (e) and when it moves from right to left. Depending on the "train", these two values need not (as shown by this example) be the same.



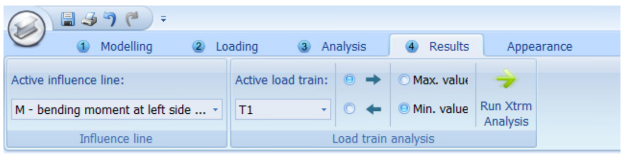
(**a**) Vertical displacement at joint 2 - max displacement: 0,40 mm



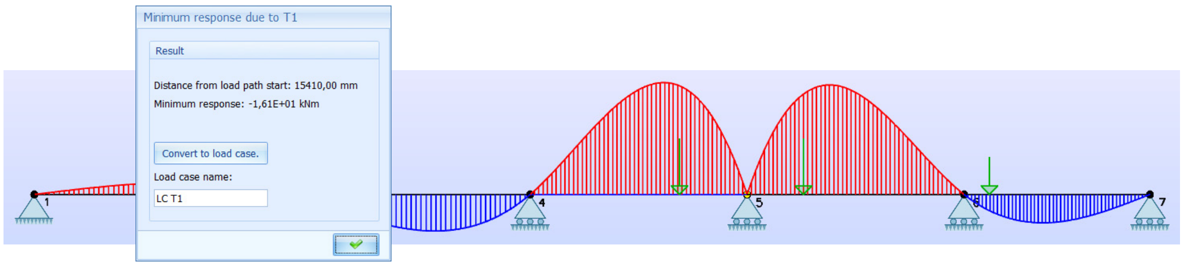
(**b**) Reaction force at joint 4 - max force: 1,00 kN



(**c**) Bending moment at joint 5 - max moment: 0,30 kNm



(**d**) Extreme value analysis



Minimum response due to T1

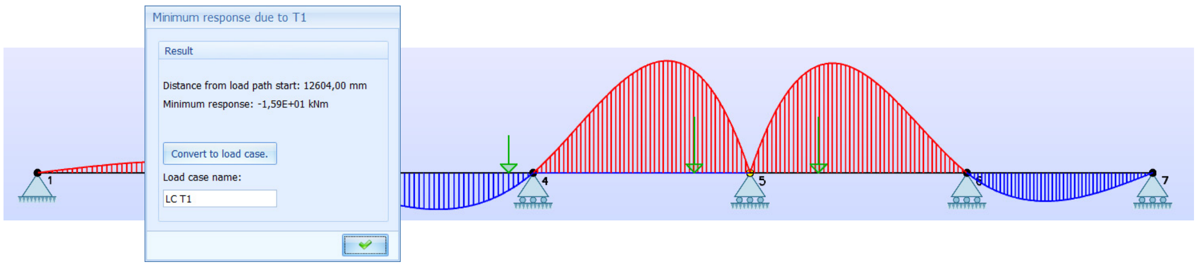
Result

Distance from load path start: 15410,00 mm
Minimum response: -1,61E+01 kNm

Convert to load case.

Load case name: LC T1

(**e**) Extreme bending moment due to train T1 moving from left to right



Minimum response due to T1

Result

Distance from load path start: 12604,00 mm
Minimum response: -1,59E+01 kNm

Convert to load case.

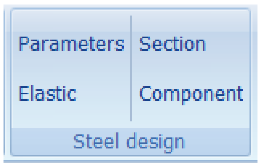
Load case name: LC T1

(**e**) Extreme bending moment due to train T1 moving from right to left

Figure 54 Influence lines and extreme value analysis

Steel design

For structures with steel components, **fap2d** offers a capacity check, according to the rules and regulations of **Eurocode 3**, for all steel members with predefined or parametric cross sections. This facility is available from the result view following a linear or nonlinear static analysis.



Three different checks are available:

- ***elastic*** *stress* control (1),
- *cross* ***section*** control (2), and
- ***component*** control (3).

1. The **elastic** capacity of a particular cross section is governed by initial yielding at the most stressed point of the section. The VON MISES yield criterion is used; hence yielding starts when the *effective* (VON MISES) stress σ_j is equal to the material yield strength f_y, or rather the design strength f_d. The code requirement is simply

$$\sigma_j \leq f_d = \frac{f_y}{\gamma_{M0}} \tag{27}$$

In 2D the effective stress, in terms of axial stress σ_x and shear stress τ, is

$$\sigma_j = \sqrt{\sigma_x^2 + 3\tau^2} \tag{28}$$

At each node of steel members, the program calculates σ_j and the capacity or utilization index κ defined as

$$\kappa = \sigma_j / f_d \tag{29}$$

and presents the index as a color "map". In order to satisfy the code κ should be smaller than or equal to 1 (one).

2. The cross **section** control is a fairly complex control which will not be dealt with in any detail here. In short the code defines 4 *classes* of cross sections, the purpose of which is to incorporate the effect of local buckling of the cross section itself on the section's capacity, in such a way as to avoid this type of buckling. Controls are carried out for the section forces present, individually and in combination, according to the rules of the code.
Sections of class 1 and 2 are checked for their plastic capacity, while sections of class 3 are subjected to an elastic control. It should be emphasized that currently *the program does not handle sections of class* 4.

3. The **component** control is applied to all straight steel beam members of constant cross section subject to compression. The in-plane buckling load is estimated through the results from a linearized buckling analysis carried out (automatically) by the program for the actual load combination. For each relevant member, the appropriate controls, considering the axial compression and the largest bending moment, are carried out, and the entire member is attributed this (highest) index.

Sections not considered are assigned class number 0, and members not considered are assigned the color white in the "capacity color map" of the structural model.

Finally it should be kept in mind that the program, being limited to a 2D world, assumes the structure to be secured (by sufficient bracing) out-of-plane. *All results apply strictly to the in-plane capacity.*

Typical results

Figure 55 shows some results from steel design checks applied to a simple steel frame

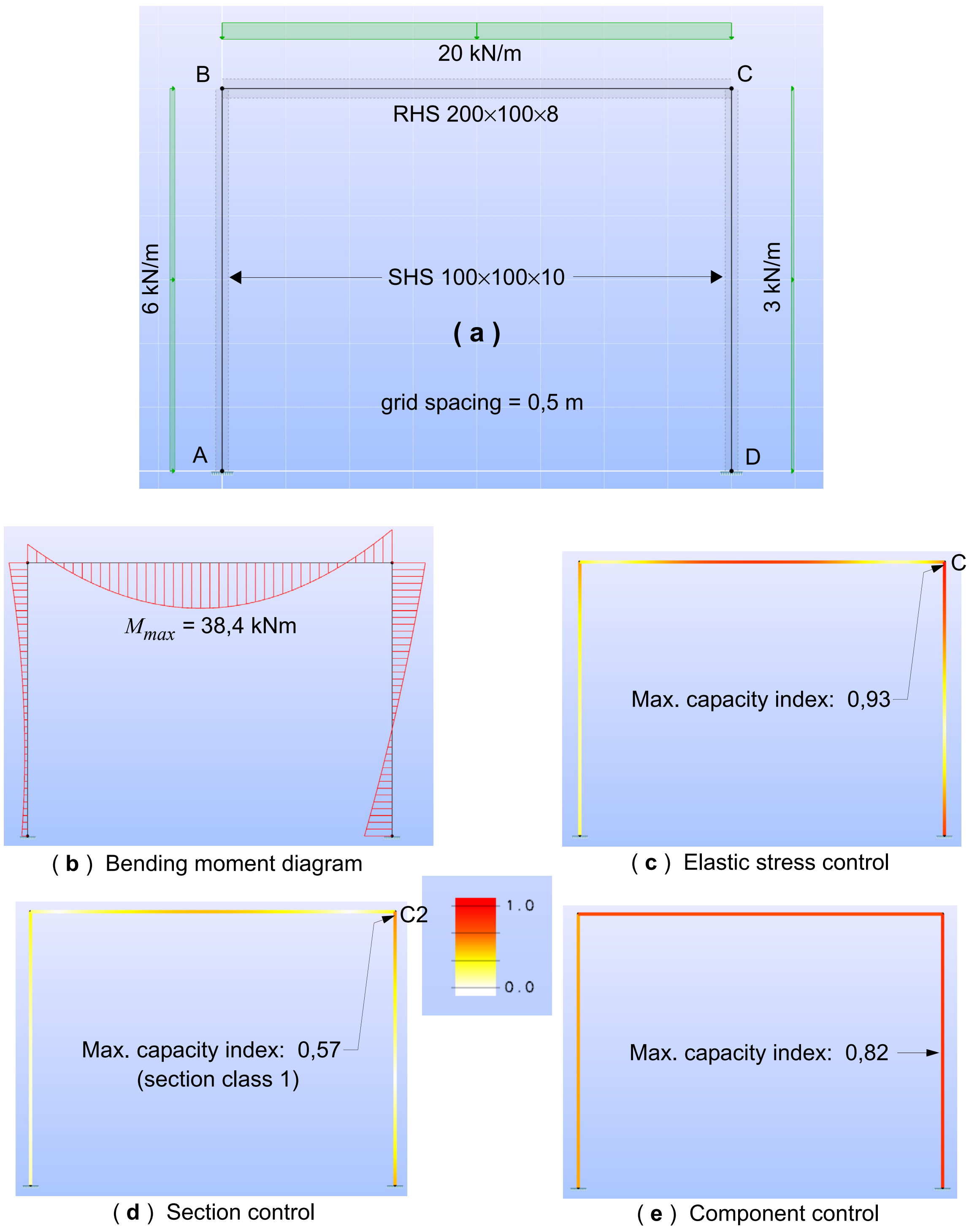
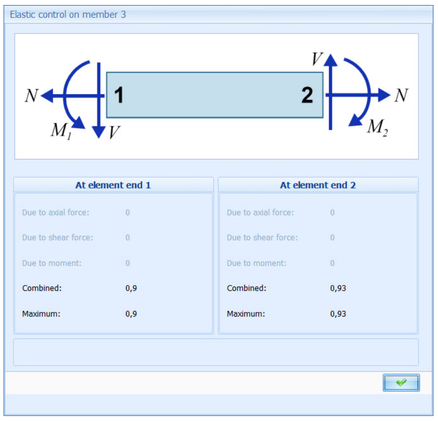


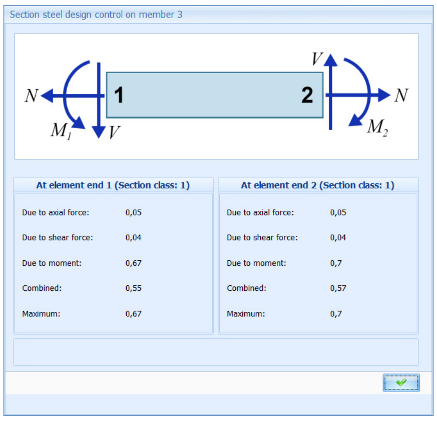
Figure 55 Examples of steel design results

following a linear static analysis. Right clicking the top of the right-hand column of figures 55c and d, produce the result boxes of figure 56a and b, respectively.


Elastic control on member 3

At element end 1		At element end 2	
Due to axial force:	0	Due to axial force:	0
Due to shear force:	0	Due to shear force:	0
Due to moment:	0	Due to moment:	0
Combined:	0,9	Combined:	0,93
Maximum:	0,9	Maximum:	0,93

(**a**) Elastic stress control


Section steel design control on member 3

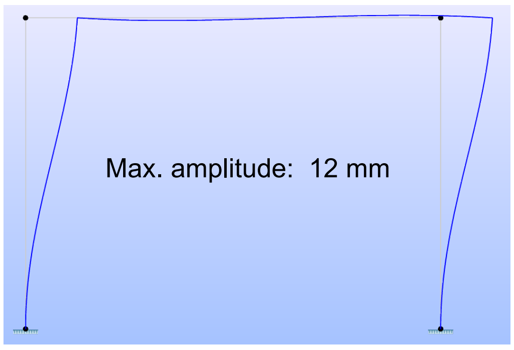
At element end 1 (Section class: 1)		At element end 2 (Section class: 1)	
Due to axial force:	0,05	Due to axial force:	0,05
Due to shear force:	0,04	Due to shear force:	0,04
Due to moment:	0,67	Due to moment:	0,7
Combined:	0,55	Combined:	0,57
Maximum:	0,67	Maximum:	0,7

(**b**) Section control

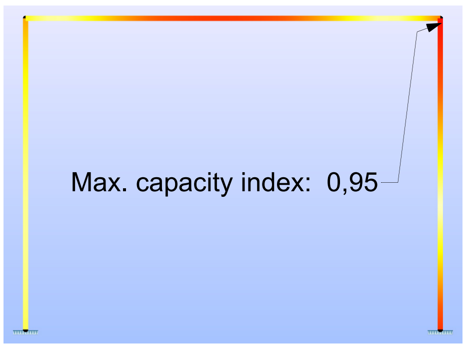
Figure 56 Detailed steel design results (apply to points C1 and C2 of figure 55)

The section control formulas of Eurocode 3 have been developed for section forces obtained by linear static analysis, and the component control following a linear analysis requires some estimate of the second order effects (*e.g.* buckling load).

If a nonlinear analysis is carried out, the computed section forces already incorporate the higher order geometric effects. Hence, the only control available, following a nonlinear analysis, is the elastic stress control. Figure 57b shows the capacity color map for the elastic stress control following a nonlinear analysis of the frame of figure 55a. The frame has, in addition to the loading of figure 55a, been subjected to a geometric imperfection, in the shape of the first buckling mode, shown in figure 57a.



(**a**) Geometric imperfection



(**b**) Elastic stress control

Figure 57 Steel design based on nonlinear static analysis (with imperfection)